



SARJANA MUDA TEKNOLOGI MAKLUMAT

**FAKULTI SAINS KOMPUTER & TEKNOLOGI MAKLUMAT
UNIVERSITI MALAYA**

**WXES 3182
LATIHAN ILMIAH TAHAP AKHIR**

Perpustakaan SKTM

3D TIC TAC TOE

**IRMA YULIANA BINTI AZMAN
WET 000165**

**PENYELIA :
PUAN NURUL FAZMIDAR MOHD NOOR**

**MODERATOR :
PUAN HANNYZURA PAL@AFFAL**

Senarai Isi Kandungan

ABSTRAK	i
PENGHARGAAN	ii
RUJUKAN	iii
 BAB 1 : PENGENALAN	
1.1 Definisi Projek	1
1.2 Objektif Projek	3
1.3 Motivasi Projek	3
1.3.1 Kenapa memilih permainan rangkaian ?	3
1.3.2 Kenapa memilih permainan 3D TIC TAC TOE ?	4
1.4 Skop Projek	5
1.5 Skop Pengguna	6
1.6 Penjadualan Projek	7
1.7 Organisasi Laporan Projek	9
 BAB 2 : KAJIAN LITERASI	
2.1 Sistem Rangkaian Komputer	11
2.1.1 Manfaat Sistem Rangkaian Komputer	13
2.1.2 Jenis – jenis Rangkaian	14

2.2	Senibina Pelanggan – Pelayan	20
2.2.1	Pelanggan – Pelayan	21
2.2.2	Komponen Pelanggan – Pelayan	22
2.3	Komunikasi Komputer	22
2.3.1	TCP / IP	24
2.3.2	Protokol Internet (IP)	26
2.4	Perbandingan Sistem Sedia Ada	27
2.4.1	Permainan Multi - Pengguna	27
2.4.2	Permainan bukan – rangkaian (non – networked)	30
2.4.3	Permainan Rangkaian	30
2.5	Bahasa Pengaturcaraan	31
2.5.1	Java	32
2.5.2	JavaScript	33
2.6	Pendekatan Pengumpulan Maklumat Projek	33
2.6.1	Kaedah dan Teknik Pengumpulan Maklumat	35

BAB 3 : METODOLOGI

3.1	Model Air Terjun	37
3.2	Fasa Umum Pembangunan Sistem	40
3.3	Model Pembangunan Yang Lain	42
3.3.1	Model Prototaip	42

BAB 4 : ANALISA SISTEM

4.1	Keperluan Sistem	46
4.1.1	Keperluan Fungsian	46
4.1.2	Keperluan Bukan Fungsian	48
4.2	Keperluan Alatan Pembangunan	50
4.2.1	Keperluan Perkakasan	50
4.2.2	Keperluan Perisian	51
4.2.3	Keperluan Peralatan Lain	52

BAB 5 : REKABENTUK SISTEM

5.1	Rekabentuk Berorientasikan Fungsian	53
5.1.1	Carta Struktur	53
5.1.2	Model Aliran Data	55
5.2	Rekabentuk Sistem Yang Dibangunkan	58
5.2.1	Rekabentuk Pelanggan – Pelayan	58
5.2.2	Rekabentuk Permainan	61
5.3	Rekabentuk Antaramuka	64
5.4	Rekabentuk Program	67
5.4.1	Pemrograman dengan Socket dan ServerSocket	67
5.4.2	Datagram Socket	68
5.4.3	Komponen Utama Dalam Pemrograman Pelanggan – Pelayan	69

BAB 6 : PENGKODAN

- 6.1 Pengenalan Pengkodan
- 6.2 Pendekatan Pengkodan
 - 6.2.1 Kaedah – Kaedah Pengkodan
 - 6.2.2 Format Pengkodan
- 6.3 Pengkodan Permainan 3D Tic Tac Toe

BAB 7 : PENYELENGGARAAN DAN PENILAIAN

- 7.1 Penyelenggaraan Sistem
 - 7.1.1 Keperluan Penyelenggaraan
 - 7.1.2 Metodologi Penyelenggaraan Sistem
 - 7.1.3 Pelan Pengembalian Bencana
- 7.2 Penilaian Sistem
 - 7.2.1 Pencapaian Objektif
 - 7.2.2 Kekuatan Sistem
 - 7.2.3 Kekangan Sistem
- 7.3 Penambahan pada masa hadapan

BAB 8 : PENGUJIAN

- 8.1 Peringkat Pengujian Sistem
 - 8.1.1 Pengujian Unit
 - 8.1.2 Pengujian Keseluruhan Sistem
 - 8.1.3 Pengujian Penerimaan

8.2	Teknik Pengujian Sistem	82
8.3	Ralat Ketika Pengujian	84
8.3.1	Ralat Kompilasi	84
8.3.2	Ralat Larian	84
8.3.3	Ralat Logik	84

BAB 9 : MASALAH DAN CADANGAN

9.1	Masalah yang dihadapi	85
9.1.1	Rujukan terhad, agak mahal dan sukar didapati	85
9.1.2	Penggunaan arahan dalam bahasa pengaturcaan Java	85
9.1.3	Pertukaran dalam fungsian sistem	86
9.2	Cadangan	86
9.2.1	Rujukan terhad, agak mahal dan sukar didapati	86
9.2.2	Penggunaan arahan dalam bahasa pengaturcaan Java	87
9.2.3	Pertukaran dalam fungsian sistem	87
9.3	Pengetahuan yang telah diperolehi	88
9.4	Kesimpulan	88

LAMPIRAN

Lampiran A – Berlawan dengan komputer	90
Lampiran B – Berlawan dengan pengguna yang lain (rangkaian)	114

Abstrak

WXES 3182 Projek Ilmiah Tahap Akhir II merupakan kursus teras fakulti bagi Fakulti Sains Komputer & Teknologi Maklumat, Universiti Malaya. Setiap pelajar yang mengambil kursus Sarjana Muda Teknologi Maklumat diwajibkan untuk mengambil subjek ini. Tujuan subjek ini adalah untuk mengaplikasikan teknik – teknik yang telah dipelajari secara teori. Pelajar dikehendaki untuk menjalankan kerja persendirian melalui satu projek yang telah ditentukan terlebih dahulu. Pelajar dikehendaki menyediakan satu system lengkap berserta dengan laporan bertulis terhadap projek yang telah dibangunkan.

Bagi projek ini, pembangunan Sistem Permainan 3D Tic Tac Toe dilaksanakan merangkumi semua aspek bagi pembangunan sesuatu sistem berkomputer. Permainan 3D Tic Tac Toe yang dibangunkan ini mempunyai konsep permainan yang sama dengan 2D Tic Tac Toe. Permainan ini diberi tajuk 3D Tic Tac Toe bersesuaian dengan 2D Tic Tac Toe yang dijadikan asas dalam penghasilan permainan ini. Perbezaan yang boleh diperolehi dari kedua – dua permainan ini adalah pada papan dimana bagi 3D Tic Tac Toe menggunakan 3 papan manakala 2D Tic Tac Toe hanya menggunakan satu papan sahaja. Permainan ini diharap dapat memberi satu dimensi baru permainan papan yang akan dimainkan secara komputer, sesuai dengan perkembangan pesat dunia pengkomputeran pada masa kini.

Penghargaan

Assalamualaikum w.b.t dan salam sejahtera

Alhamdulillah, bersyukur saya ke hadrat Ilahi kerana dengan limpah kurnianya dapat saya menyiapkan laporan Projek Ilmiah Tahap Akhir II ini. Di kesempatan ini, ingin saya merakamkan ucapan ribuan terima kasih kepada Puan Nurul Fazmidar Mohd Noor selaku penyelia dan juga kepada Puan Hannyzzura Pal @ Affal selaku moderator bagi projek ini, segala sokongan dan bimbingan yang diberikan amatlah dihargai.

Seikhlas penghargaan juga saya tujukan kepada kedua ibu bapa dan seisi keluarga iaitu Encik Azman, Puan Siti Zahrah, Amelia, Noraishah, Norsyazwani dan Safian di atas dorongan dan sokongan yang diberikan. Tidak dilupakan kepada rakan – rakan seperjuangan – Sufyan, Emyzai, Nik, Diana, Mazrita, Fauzana, Dilla, Maryam, Ain, Baizura, Fizoh yang sama – sama bertungkus lumus dalam menyiapkan laporan projek ini. Terima kasih di atas sumbangan masa dan idea dari kalian semua. Tidak ketinggalan kepada semua pensyarah yang telah menyampaikan ilmunya kepada saya sepanjang saya berada di Universiti Malaya ini dan juga kepada semua pihak yang terlibat secara langsung atau tidak langsung sepanjang laporan Projek Ilmiah Tahap Akhir II ini disiapkan.

Sekian, terima kasih.

IRMA YULIANA AZMAN

Jabatan Multimedia (Teknologi Maklumat),
Fakulti Sains Komputer & Teknologi Maklumat,
Universiti Malaya.

1.1. Latar Belakang

Bab 1: Pengenalan

University of Malaysia

Bab 1 : Pengenalan

1.1 Definisi Projek

Penggunaan komputer pada masa kini telah menjadi seperti sebahagian dari hidup kita. Perjalanan dan tugas seharian menjadi semakin mudah dan tanpa disedari kita bergantung kepada mesin yang kita panggil komputer ini. Begitu juga dengan tanggungjawab dan beban kerja di pejabat, di mana maklumat – maklumat perlulah dihantar dengan cepat dan pantas bagi mengelakkan berlakunya kerugian atau sebarang kebocoran maklumat. Contohnya, ahli – ahli perniagaan, penghantaran maklumat yang pantas dan selamat berupaya menghasilkan keuntungan yang berlipat ganda.

Di bidang pendidikan pula, gabungan di antara penggunaan komputer dengan teknologi telekomunikasi telah menghasilkan apa yang kita kenali sebagai Teknologi Maklumat. Teknologi Maklumat telah berperanan sebagai asas untuk menyimpan maklumat, mengolah maklumat, menyalur dan mengedarkan maklumat di dalam pelbagai bentuk dan cara. Bahan atau maklumat tersebut dapat disalurkan sama ada di dalam bentuk teks, gambar, grafik, suara, animasi dan video atau gabungan kesemua unsur secara interaktif yang juga dikenali sebagai Multimedia.

Dewasa kini, kita dapat melihat pelbagai jenis permainan komputer di pasaran. Kebanyakan permainan tersebut merupakan permainan yang boleh dimainkan secara manual, namun pengguna sekarang lebih gemar menggunakan permainan komputer. Permainan komputer boleh dikategorikan sebagai sebuah permainan yang tidak statik. Ianya berubah mengikut tindakan yang dilakukan oleh pemain itu sendiri. Kebanyakan permainan ini mempunyai darjah penyelesaian teka – teki.

Permainan komputer adalah interaktif dan mempunyai matlamatnya yang tersendiri. Tahap permainan berubah mengikut tindakan yang dilakukan oleh pemain itu sendiri, makin cergas pemain berfikir, makin sukar pihak lawan memenangi permainan tersebut. Terdapat pelbagai cara boleh dilakukan oleh pengguna untuk memerangkap pihak lawan di dalam sesuatu permainan iaitu dengan memikirkan beberapa strategi serta taktik – taktik yang berkesan.

Dengan berkembangnya teknologi, komputer – komputer boleh digabungkan dalam satu rangkaian komputer di mana ianya membolehkan pengguna berhubung antara satu sama lain dan ini sekaligus menjadikan permainan komputer sebagai salah satu permainan rangkaian. Permainan komputer berkonsepkan rangkaian lebih digemari oleh pengguna kerana ianya merupakan permainan yang interaktif dan mencabar. Dunia permainan komputer begitu meluas ketika ini, ianya boleh dikelaskan kepada beberapa kategori seperti permainan menguji minda, strategi, aksi, kerjasama antara pemain, persaingan sesama pengguna dan lain – lain. Permainan komputer yang telah sedia ada adalah seperti Tetris, Foxy Columns, Pacman, Chess, FIFA 2002 dan sebagainya.

Perkara utama yang perlu dipertimbangkan di dalam membangunkan projek ini adalah konsep hubungan pelanggan – pelayan dimana ia menjadi aset utama dalam memastikan permainan ini berfungsi. Pemain juga boleh berada dimana – mana sahaja samada pengguna bersebelahan ataupun pengguna pada bilik yang lain ataupun di mana sahaja asalkan mereka dihubungkan dengan rangkaian komputer. Kotak perbualan yang disediakan menjadi pengantara bagi setiap pengguna berinteraksi.

1.2 Objektif Projek

Di antara objektif dalam melaksanakan projek ini adalah :

- i. Untuk mencipta permainan yang membolehkan pengguna bermain dengan komputer ataupun dengan pengguna lain melalui rangkaian komputer.
- ii. Untuk mencipta permainan yang membenarkan pengguna berinteraksi antara satu sama lain.
- iii. Menyediakan antaramuka permainan yang interaktif, mudah digunakan dan ramah pengguna.
- iv. Mengenalpasti teknologi rangkaian yang digunakan dalam membangunkan permainan ini.

1.3 Motivasi Projek

Bahagian ini akan memfokus kepada pemilihan teknikal yang diambil dalam membangunkan projek permainan 3D Tic Tac Toe.

1.3.1 Kenapa memilih permainan rangkaian ?

Permainan rangkaian menyediakan pelbagai elemen – elemen yang lebih interaktif berbanding permainan komputer yang hanya berdiri sendiri sahaja. Pada komputer yang berdiri sendiri, kita hanya boleh melawan komputer atau maksimum dua orang dengan menggunakan kayu bedik (joystick) pada sebuah komputer sahaja. Dengan sistem rangkaian komputer, pengguna boleh bermain lebih dari tiga orang dengan menggunakan komputer masing – masing tanpa perlu berkongsi komputer.

Pengguna juga boleh memilih rakan – rakan yang dikenali dalam memainkan permainan komputer berkonsep rangkaian ini. Melalui kemajuan teknologi rangkaian internet, ribuan pengguna boleh bermain pada satu komputer pelayan. Pengguna boleh berinteraksi dengan lebih jauh lagi dan bukan hanya terhad kepada satu kawasan sahaja seperti berinteraksi dengan pengguna yang berada di Amerika Syarikat, Jepun, Indonesia dan sebagainya. Secara tidak langsung, pengguna dapat berinteraksi dengan pelbagai jenis lapisan pengguna ketika dalam permainan.

1.3.2 Kenapa memilih permainan 3D Tic Tac Toe

Permainan 3D Tic Tac Toe adalah terbitan daripada permainan 2D Tic Tac Toe. Ianya dipilih kerana permainan ini merupakan sebuah permainan yang agak mencabar serta menguji minda pengguna di samping memberikan hiburan kepada pengguna pada masa lapang.

Mungkin pengguna sedar akan kewujudan permainan ini tetapi tidak mengetahui bagaimana permainan ini dimainkan, oleh itu ianya dipilih bagi memperkenalkan dengan lebih dekat lagi tentang konsep permainan 3D Tic Tac Toe kepada pengguna. Permainan 2D Tic Tac Toe adalah merupakan sebuah permainan klasik yang telah wujud pada awalan permainan komputer dibangunkan, ianya merupakan sebuah permainan yang sudah biasa dan dikenali di kalangan pengguna. Permainan ini agak membosankan kerana ianya mudah untuk dimenangi dan tidak mencabar. Pengguna disediakan dengan sebuah papan yang mempunyai 9 kotak dan juga dua bentuk nod iaitu bulat dan pangkah. Untuk memenangi permainan, pengguna perlu meletakkan nod pada barisan samada dalam keadaan melintang ataupun mengufuk. Masa untuk

permainan ini juga adalah terlalu cepat dan persekitaran yang disediakan tidak begitu menarik.

Oleh sebab itu, permainan 2D Tic Tac Toe telah diubah kepada permainan 3D Tic Tac Toe di mana ianya agak berbeza dengan permainan klasik tersebut. Di dalam permainan 3D Tic Tac Toe, pengguna disediakan 3 papan yang mempunyai 27 kotak serta dua buah nod iaitu bulat dan pangkah, bagi memenangi permainan ini pengguna perlu meletakkan nod pada satu barisan pada keadaan melintang, menegak ataupun pepenjuru. Jadi dapat dibezakan di sini di antara permainan klasik 2D Tic Tac Toe dan juga 3D Tic Tac Toe, iaitu 3 papan dan juga pepenjuru. Ini menyebabkan permainan yang akan dibangunkan agak sukar untuk dimenangi dan mengambil masa pengguna memikirkan cara bagaimana untuk mengalahkan pihak lawan. Ianya adalah bersesuaian dengan kehendak pengguna sekarang iaitu corak permainan yang mencabar minda dan semakin kompleks.

1.4 Skop Projek

Skop projek permainan 3D Tic Tac Toe meliputi beberapa aplikasi yang disediakan untuk pengguna. Aplikasi – aplikasi ini akan dimuatkan ke dalam beberapa modul. Di antara modul – modul yang akan digunakan dalam membangunkan projek ini adalah modul permainan, modul pelanggan – pelayan (client / server) dan juga modul kotak perbualan. Pada setiap modul mempunyai fungsinya tersendiri, bagi modul permainan dapat dinyatakan mengenai bagaimana permainan itu berjalan dan elemen – elemen yang digunakan dalam persekitaran permainan tersebut. Pemilihan kepada bilangan pengguna yang akan memainkan permainan itu, iaitu samada ingin bermain

perseorangan ataupun bermain dengan pengguna yang lain. Bagi permainan perseorangan, pengguna akan berlawan dengan komputer itu sendiri. Modul permainan ini juga akan memaparkan keseluruhan antaramuka pengguna yang direkabentuk mengikut keperluan pengguna.

Bagi modul pelanggan – pelayan pula ianya dianggap sebagai modul yang paling utama dan penting dalam membangunkan projek permainan 3D Tic Tac Toe ini. Ini adalah kerana permainan komputer yang akan dibangunkan ini dihubungkan melalui senibina pelanggan – pelayan itu sendiri. Pada hakikatnya, rekabentuk bagi permainan ini boleh dibahagikan kepada dua komponen iaitu pelanggan (client) dan juga pelayan (server). Kedua – dua komponen ini secara logiknya adalah berasingan, ianya dihubungkan menerusi protokol pada permainan tersebut dan juga protokol yang telah ditetapkan pada komputer pelanggan serta komputer pelayan. Fungsi pelayan di dalam permainan ini adalah sama seperti seorang pengadil dimana ianya mengurus dan mengawal segala gerak geri permainan di samping membantu pemain berkomunikasi antara satu sama lain dengan efektif dan berkesan. Selain itu, ia juga mengawal aliran permainan tersebut dengan memberi maklumat terkini mengenai status setiap pemain di dalam permainan 3D Tic Tac Toe. Status pemain akan sentiasa berubah dan diperbaharui mengikut giliran masing – masing. Oleh itu, data yang diterima pada setiap pemain adalah yang terkini dan juga tepat.

Bagi komputer pelanggan pula, tugasnya adalah menghantar maklumat seperti login nama pemain dan menghantar maklumat pergerakan pengguna semasa permainan diaktifkan oleh pelayan. Selain daripada itu, pelanggan juga bertanggungjawab memaparkan grafik antaramuka permainan dan interaksi antara pengguna akan berlaku

pada komputer pelanggan. Perkara penting yang perlu dititikberatkan dalam permainan rangkaian adalah pelayan, di mana pelayan haruslah sentiasa berfungsi pada setiap masa.

Modul yang terakhir adalah modul kotak perbualan di mana di dalam modul ini menerangkan tentang elemen – elemen yang terdapat dalam kotak perbualan tersebut. Untuk mengelakkan daripada berlakunya sebarang perbualan yang tidak diingini maka konsep perbualan terkawal telah dipilih dalam modul kotak perbualan ini. Pengguna hanya boleh berinteraksi dengan menggunakan kata – kata yang telah disediakan oleh pembangun sistem dan bukan menulis dengan ayat sendiri. Terdapat juga ikon yang memaparkan pelbagai jenis reaksi muka dan juga objek – objek tertentu.

1.5 Skop Pengguna

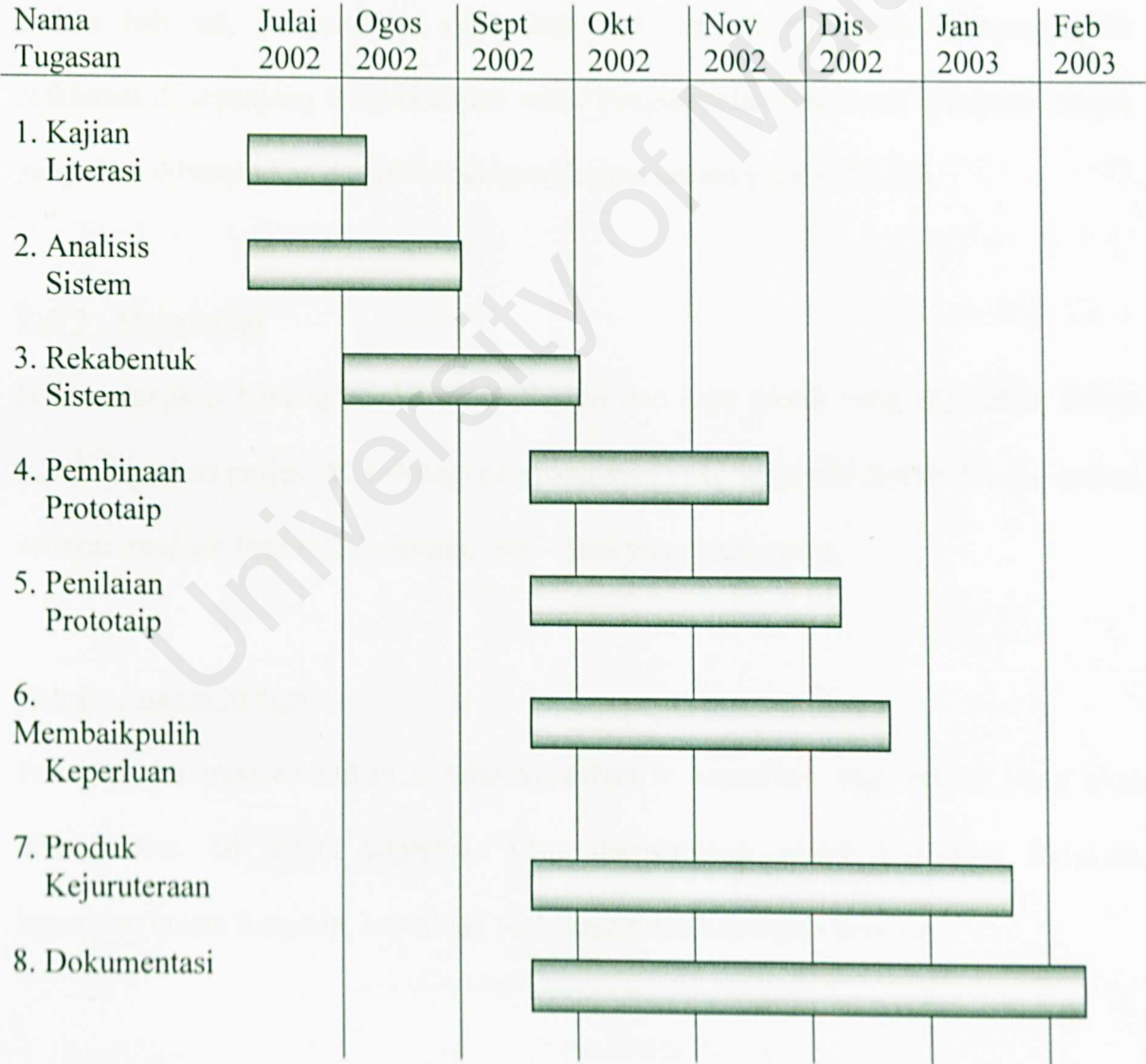
Sesiapa sahaja boleh bermain permainan 3D Tic Tac Toe. Reka bentuk permainan ini adalah mudah dan senang untuk digunakan. Ciri permainan ini juga boleh diterima oleh semua peringkat umur dan ianya mudah untuk difahami. Pemain hanya perlu mengetahui cara – cara permainan dan juga peraturannya sahaja. Selepas permainan dimulakan, terpulang kepada pemain untuk memikirkan strategi bagi memerangkap pihak lawan dan sekaligus memenangi permainan 3D Tic Tac Toe. Permainan ini sesuai diperkenalkan kepada pengguna berumur 7 tahun ke atas.

1.6 Penjadualan Projek

Penjadualan projek melibatkan pembahagian kerja kepada aktiviti berasingan dan membuat anggaran masa dan sumber yang akan digunakan untuk menyiapkan

setiap aktiviti. Biasanya sebahagian daripada aktiviti ini adalah aktiviti bebas yang boleh dilaksanakan serentak, manakala yang lain pula adalah aktiviti sandaran yang hanya boleh dilaksanakan apabila aktiviti tertentu telah siap.

Dalam persekitaran projek, penjadualan yang baik adalah amat penting. Ianya bagi mengelakkan daripada permasalahan aktiviti – aktiviti projek bertanggung dari masa ke semasa dan projek tidak dapat disiapkan pada masa yang telah ditetapkan. Carta Gantt di bawah menunjukkan penjadualan projek, di mana setiap aktiviti diwakili satu bar yang dilukis sepanjang tarikh mula dan tamat dalam penjadualan projek :



1.7 Organisasi Laporan Projek

Laporan projek ini mengandungi beberapa bab di mana setiap bab mempunyai penerangan mengenai aktiviti – aktiviti yang akan dilaksanakan :

Bab 1 : Pengenalan

Bab ini menerangkan pengenalan secara ringkas mengenai projek yang hendak dilaksanakan dan perancangan untuk projek yang akan dibangunkan.

Bab 2 : Kajian Literasi

Dalam bab ini, penyelidikan dijalankan terhadap sistem untuk mengumpulkan maklumat di sepanjang tempoh kajian awal. Pengumpulan maklumat mengenai projek yang akan dibangunkan dan perbandingan dengan sistem yang sedia ada.

Bab 3 : Metodologi

Ia menerangkan tentang kaedah penyelidikan dan juga teknik yang digunakan dalam membangunkan projek. Metodologi pembangunan yang sedia ada dipilih dan digunakan sebagai panduan bagi melaksanakan fasa – fasa yang berikutnya.

Bab 4 : Analisa Sistem

Bahagian ini menyenaraikan senarai keperluan – keperluan bagi projek yang akan dibangunkan. Di antara keperluan yang disenaraikan adalah keperluan fungsian, keperluan bukan fungsian, keperluan perkakasan dan keperluan perisian.

Bab 5 : Rekabentuk Sistem

Di dalam bab ini, menerangkan tentang rekabentuk antaramuka yang telah direka, aliran maklumat dan juga modul – modul yang terlibat dalam projek yang akan dibangunkan.

Bab 6 : Perlaksanaan / Pembangunan Sistem

Bab ini akan menerangkan mengenai bagaimana sistem dibangunkan, himpunan daripada analisa sistem, modul – modul dan juga rekabentuk yang telah direka.

Bab 7 : Pengujian Sistem

Bahagian pengujian sistem akan menerangkan kepada pengguna mengenai objektif pengujian sistem, peringkat – peringkat ujian yang terlibat dan juga teknik pengujian sistem yang dibangunkan. Di samping itu juga, ianya bagi mengesahkan sama ada sistem berfungsi mengikut keperluan dan spesifikasi yang telah ditentukan.

Bab 8 : Perbincangan

Bab ini akan membincangkan masalah yang timbul ketika penghasilan laporan dibuat serta penyelesaiannya, keputusan yang diperolehi daripada penghasilan sistem yang dibangunkan, kelebihan dan juga kelemahan sistem, peningkatan terhadap sistem pada masa hadapan, menggunakan cadangan dan juga membuat kesimpulan terhadap projek yang dijalankan.

Bab 2: Kajian Literasi

Bab 2 : Kajian Literasi

Bab ini menyenaraikan kajian yang telah dilaksanakan dalam membantu permasalahan sebelum projek dapat dilaksanakan. Ianya meliputi kajian serta analisa sistem yang telah sedia ada, kajian berkenaan teknik serta aplikasi yang akan digunakan.

2.1 Sistem Rangkaian Komputer

Sistem rangkaian komputer (Networking) turut memainkan peranan yang tidak kurang pentingnya di dalam era teknologi maklumat terkini. Sistem rangkaian komputer merupakan sistem komputer yang berpaut bersama dengan terminal dan peralatan perisian yang lain melalui talian komunikasi. Rangkaian membenarkan pengguna yang berada pada kedudukan taburan geografi yang pelbagai untuk berkongsi aturcara, data dan peralatan. Teknologi rangkaian komputer ini yang pada awalnya hanya digunakan di kalangan masyarakat teknologi serta mereka yang gemar mencuba teknologi yang baru, tetapi kini menjadi antara teknologi yang mendapat penerimaan luas di kalangan masyarakat.

Tidak keterlaluan jika kita mengatakan teknologi rangkaian sebenarnya membuka suatu dimensi baru di dalam penggunaan komputer peribadi. Pengguna yang baru mengenali teknologi rangkaian seringkali tertanya-tanya mengenai kelainan dan ciri perbezaan yang dibawa oleh teknologi rangkaian kepada dunia perkomputeran. Sebelum teknologi rangkaian diperkenalkan serta digunakan secara meluas, kebanyakan komputer peribadi lebih berfungsi sebagai sebuah komputer tunggal. Pertamanya, ia tidak dirangkaian iaitu disambungkan dengan mana-mana komputer yang lain. Oleh yang demikian, komputer tersebut hanya berupaya menggunakan sumber yang

disediakan serta terdapat pada komputer tersebut sahaja. Dalam erti kata lain, pengguna tersebut hanya dapat menggunakan perisian serta perkakasan yang terdapat serta disambungkan kepada komputer tersebut sahaja. Sebelum wujudnya sistem rangkaian komputer, pengguna yang ingin berkongsi perisian ataupun fail di dalam persekitaran komputer tunggal terpaksa menggunakan media storan yang boleh dialihkan seperti disket. Namun begitu, kehadiran teknologi rangkaian komputer membolehkan apa yang suatu ketika dahulu dianggap mustahil menjadi kenyataan yang menguntungkan, menyeronokkan dan merealisasikan impian ramai pengguna.

Sistem rangkaian komputer merupakan satu sistem komputer yang mengandungi dua atau lebih komputer yang dihubungkan di antara satu sama lain melalui sistem pengoperasian yang selaras walaupun komputer-komputer tersebut berada di tempat-tempat yang berbeza. Rangkaian pula adalah suatu set peranti (dipanggil nod) yang dihubungkan oleh media penghantaran (dikenali sebagai *pautan*). Pautan juga dipanggil sebagai *saluran komunikasi*. Nod boleh terdiri daripada komputer kerangka utama, komputer peribadi, terminal, pencetak atau peranti lain yang berupaya untuk menghantar atau menerima data.

Rangkaian membolehkan pemprosesan teragih dijalankan, di mana sesuatu tugas itu boleh dibahagi kepada berbilang komputer. Dua sebab utama mengapa rangkaian diperlukan, pertama adalah untuk berkongsi peralatan dan perkakasan dan kedua pula menjadikan kita lebih produktif, kerana kita boleh berkongsi maklumat juga peralatan tanpa perlu meninggalkan pejabat atau kediaman masing – masing. Rangkaian menghubungkan pengguna dari satu bilik ke satu bilik yang lain, dalam satu bangunan, berlainan bangunan ataupun berlainan negara.

2.1.1 Manfaat sistem rangkaian komputer

Wujudnya sistem rangkaian komputer telah memberikan beberapa manfaat yang besar kepada pengguna – penggunanya. Melalui rangkaian ini, pengguna boleh berkongsi menggunakan perisian serta fail. Perkongsian , penyalinan serta penghantaran perisian serta fail dapat dilaksanakan secara talian. Ini bermakna pengguna tidak perlu menghabiskan sumber kewangan yang besar untuk membelikan perisian untuk setiap unit komputer mereka. Pengguna hanya perlu membeli sebuah perisian versi rangkaian contohnya LAN Version sudah memadai untuk semua komputer yang terdapat di dalam rangkaian tersebut menggunakannya.

Pengguna juga boleh berkongsi data atau perisian pada masa yang serentak contohnya tiga orang pekerja di dalam suatu jabatan, bekerja dengan menggunakan hamparan helaian yang sama. Keupayaan ini khususnya berguna di dalam kes pangkalan data, yang perlu dicapai dan ditukar oleh beberapa orang di dalam satu organisasi. Selain daripada itu ialah kemampuan para pengguna rangkaian berhubung serta berkomunikasi di antara satu sama lain. Pengguna juga boleh bertukar-tukar mesej atau pesanan melalui komputer melalui proses yang dikenali sebagai mel elektronik serta perisian sembang (chat) di dalam persekitaran rangkaian dimana ia juga membolehkan sesebuah pejabat serta tempat kerja berkomunikasi serta berhubung di antara satu sama lain. Malah, penghantaran memo serta dokumen di dalam sesebuah syarikat yang kelazimannya merupakan budaya serta kaedah kerja di dalam sesebuah pejabat atau syarikat sudah menjadi tidak relevan serta berguna lagi.

Sistem rangkaian komputer turut membenarkan beberapa komputer berkongsi perkakasan yang sama contohnya mesin pencetak, pengimbas, CD ROM dan

kelengkapan faks yang mahal. Contohnya, sebuah pejabat yang mempunyai dua puluh buah komputer, tidak perlu mendapatkan dua puluh buah mesin pencetak untuk membolehkan mereka melakukan tugas cetakan mereka. Sebuah mesin pencetak yang disambungkan kepada mana-mana komputer peribadi di dalam rangkaian tersebut akan dapat dikongsi bersama oleh komputer-komputer peribadi di dalam sistem rangkaian tersebut.

Konsep rangkaian juga tidak terhad kepada satu-satu lokasi atau perniagaan sahaja. Internet sebagai contoh, satu daripada beberapa rangkaian yang ada, berupaya menghubungkan pengguna komputer dari merata pelusuk dunia. Pengguna boleh menjalankan atur cara komputer dari benua lain dan boleh mencipta halaman rerumah (homepage) supaya berjuta-juta orang yang tidak dikenali boleh membacanya. Internet juga membolehkan pelajar-pelajar khasnya serta pengguna - pengguna amnya mencari pelbagai variasi maklumat yang mereka inginkan mengenai apa - apa sahaja dari mana - mana sudut dunia dengan hanya melayari dunia internet. Satu kelebihan lagi ialah segala maklumat yang mereka inginkan itu akan mereka dapati di dalam jangkamasa yang singkat sahaja.

2.1.2 Jenis – jenis rangkaian

Jenis – jenis rangkaian sebenarnya ditakrifkan mengikut lindungan geografi sesuatu rangkaian itu. Selain daripada itu, perbezaan di antara skema rangkaian – rangkaian tersebut melibatkan jarak dan orientasi. Biasanya, ia berorientasikan kerangka pusat dan menggunakan talian telefon atau bergantung kepada geganti gelombang mikro untuk menghantar data.

Jenis – jenis rangkaian

Jenis – jenis rangkaian sebenarnya ditakrifkan mengikut lingkungan geografi sesuatu rangkaian itu. Selain daripada itu, perbezaan di antara skema rangkaian-rangkaian tersebut turut melibatkan jarak dan orientasi. Biasanya, ia berorientasikan kerangka pusat dan menggunakan talian telefon atau bergantung kepada geganti gelombang mikro untuk menghantar data. Terdapat tiga jenis rangkaian komputer yang asas iaitu ; Rangkaian Kawasan Setempat (Local Area Network – LAN), Rangkaian Kawasan Metropolitan (Metropolitan Area Network – MAN) dan Rangkaian Kawasan Luas (Wide Area Network – WAN).

RANGKAIAN KAWASAN SETEMPAT (LAN)

Rangkaian jenis LAN ini biasanya terhad kepada batasan geografi yang kecil misalnya rangkaian yang meliputi sesebuah bangunan, sekolah, pejabat atau kampus. Fungsi asas LAN sebagai satu rangkaian yang menghubungkan sebilangan komputer di dalam kawasan yang terhad seperti di dalam kampus universiti, di dalam sesebuah bangunan dan di dalam bilik serta kerap kali dihubungkan dengan komputer kerangka utama atau minikomputer.

Namun begitu, jarang sekali rangkaian LAN meliputi kawasan yang lebih daripada satu batu jaraknya. Komputer-komputer ini dapat dihubungkan dengan pelbagai kaedah seperti kabel pasangan terpiuh atau berpintal, gentian optik, talian telefon dan cahaya inframerah serta isyarat radio. Situasi yang paling biasa ditemui oleh

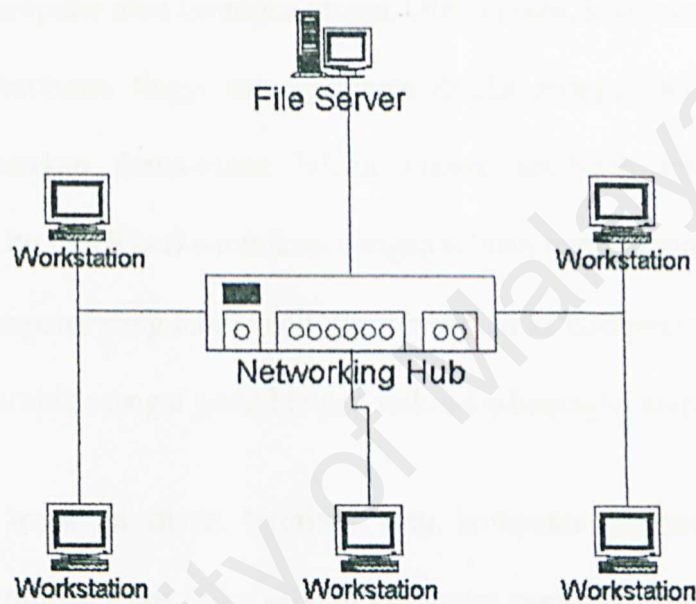
rangkaian ialah mesej dari satu komputer ke satu komputer yang lain atau arahan supaya menjalankan atur cara yang tersimpan di dalam rangkaian. Data atau atur cara yang diminta oleh mesej mungkin disimpan pada komputer yang digunakan oleh seseorang rakan sekerja dalam rangkaian atau pada pelayan fail (server), suatu komputer khusus.

Pelayan fail biasanya adalah merupakan komputer berkuasa tinggi dan mempunyai pemacu cakera keras yang tidak digunakan secara eksklusif oleh individu pada rangkaian itu serta cakera padat (CD drive). Pelayan fail juga seharusnya mempunyai storan semua perisian yang mengawal pengurusan sesuatu sistem rangkaian itu. Selain daripada itu, pelayan fail wujud hanya untuk melayan semua komputer lain yang berada di dalam sistem rangkaian tersebut. Komputer-komputer yang disambung kepada pelayan fail dikenali sebagai stesen kerja (workstations). Stesen kerja tidak semestinya berkuasa tinggi seperti pelayan komputer namun begitu, komputer-komputer tersebut boleh mempunyai perisian tambahan pada cakera keras masing-masing.

Manakala, setiap satu komputer yang menggunakan khidmat pelayan fail dikenali sebagai pelanggan (client). Pencetak boleh juga disambung kepada rangkaian supaya ia boleh dicapai oleh semua pelanggan LAN. Untuk menggunakannya, rangkaian mestilah menerima permintaan daripada komputer individu atau nod yang dihubungkan pada rangkaian dan ia mestilah mempunyai cara untuk mengendalikan permintaan serentak bagi perkhidmatannya. Sebaik sahaja komputer mendapat perkhidmatan rangkaian, ia haruslah mempunyai cara untuk menghantar mesej dari satu

komputer kepada komputer yang lain supaya ia hanya menuju kepada nod yang diperlukan dan tidak muncul di komputer yang lain. Rangkaian pula mestilah melakukan semua ini secepat mungkin sambil memberi perkhidmatannya secara serata mungkin mengikut giliran di antara nod lain di dalam LAN.

Basic LAN Topology



Rajah 2.1 : Rangkaian Kawasan Setempat (LAN)

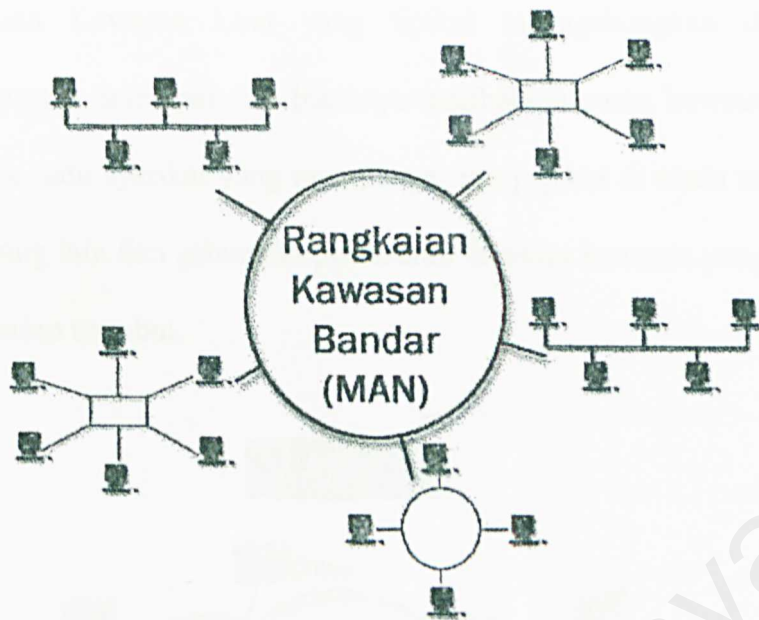
RANGKAIAN KAWASAN METROPOLITAN (MAN)

Rangkaian Kawasan Metropolitan (Metropolitan Area Network-MAN) lazimnya meliputi suatu kawasan geografi yang agak luas berbanding dengan rangkaian yang diliputi oleh LAN. Rangkaian MAN biasanya beroperasi di bandar-bandar, gabungan beberapa buah sekolah ataupun di sesebuah daerah. Dengan menjalankan suatu rangkaian perhubungan yang sederhana besarnya di dalam lingkungan susun atur

geografi yang besar, informasi dan maklumat dapat disebarkan dengan meluas, cepat dan berkesan. Perpustakaan – perpustakaan awam dan agensi-agensi kerajaan biasanya gemar menggunakan MAN untuk berhubung dengan warga setempat dan pihak pengurusan sektor swasta.

Di dalam bidang perniagaan contohnya, satu komputer peribadi menghantar data kepada satu minikomputer atau kerangka utama. Oleh kerana, komputer-komputer yang lebih besar atau berkuasa tinggi sememangnya direka sebagai pelayan (server), ia berfungsi membenarkan mana-mana laluan masuk sesebuah terminal, sesebuah komputer peribadi itu boleh berkomunikasi dengan sebuah minikomputer atau kerangka utama. Apabila komputer yang lebih kecil disambungkan ke komputer yang lebih besar, perhubungannya dirujuk sebagai perhubungan mikro-ke-kerangka utama.

Komputer besar di mana terminal atau komputer peribadi disambungkan dikenali sebagai komputer hos. Jika sebuah komputer peribadi itu digunakan sebagai terminal, perisian memindahkan fail membenarkan pengguna untuk memuat-turun fail dari hos ataupun memuat-naik data. Memuat-turun fail bermaksud membuka dan mengambil data dari sebuah komputer peribadi yang lain dan menghantar data ke komputer yang berkenaan yang diminta oleh pengguna. Memuat-naik fail pula bererti pengguna membuka fail data dan menghantarkannya ke sebuah komputer yang lain.

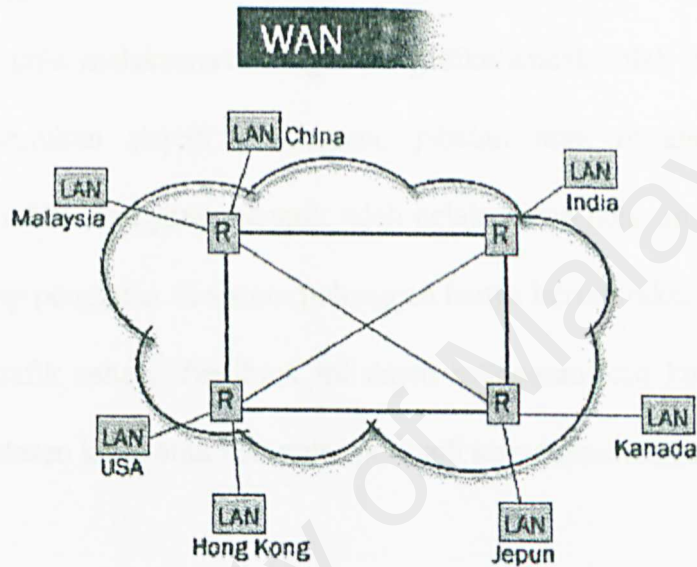


Rajah 2.2 : Rangkaian kawasan metropolitan (MAN)

RANGKAIAN KAWASAN LUAS (WAN)

Rangkaian Kawasan Luas (Wide Area Network – WAN) menghubungkan suatu kawasan yang lebih luas dari segi geografinya, contohnya menghubungkan Florida, Amerika Syarikat dengan dunia. Rangkaian WAN berupaya menghubungkan sekolah-sekolah di Florida dengan tempat-tempat lain di dunia contohnya Tokyo dalam hanya beberapa minit sahaja, tanpa perlu menyediakan sejumlah wang yang besar untuk membayar bil telefon. Namun begitu, rangkaian WAN ini agak rumit dan kompleks. Ia memerlukan pelbagai perkakasan dan perisian sebelum membolehkan rangkaian setempat dan metropolitan berhubung dengan komunikasi secara global dan antarabangsa seperti Internet. Secara lahirnya, rangkaian WAN kelihatan hampir serupa dan tidak banyak bezanya dengan LAN atau MAN.

Rangkaian Kawasan Luas yang tipikal menggabungkan dua atau lebih Rangkaian Kawasan Setempat dan biasanya melibatkan suatu kawasan geografi yang luas. Contohnya, satu syarikat yang mempunyai ibu pejabat di suatu tempat, kilang di suatu tempat yang lain dan gabungan pemasaran di suatu kawasan yang agak jauh dari kedua-dua kawasan tersebut.



Rajah 2.3 : Rangkaian kawasan luas (WAN)

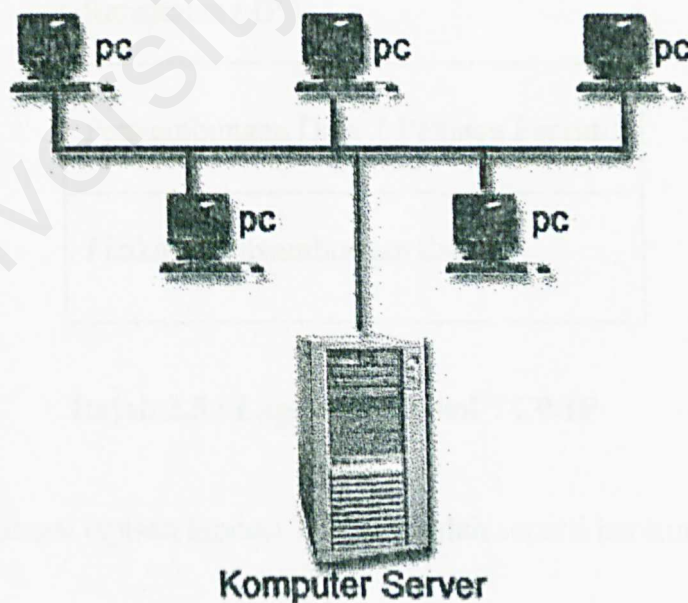
2.2 Senibina Pelanggan – Pelayan

Secara khusus, pertukaran persekitaran teknologi komputer dan telekomunikasi telah dan akan berlaku ialah perubahan dari era persekitaran pengkomputeran hos kepada persekitaran rangkaian. Persekitaran baru ini mengelolakan teknologi, data dan aplikasi perisian dalam bentuk bersepadu. Ia juga dikenali sebagai persekitaran pelanggan - pelayan yang membolehkan penjimatan kos, lebih berkuasa dalam daya pemprosesan dan menyediakan platform pengkomputeran dinamik.

2.2.1 Pelanggan – Pelayan

Model pelanggan - pelayan boleh dibahagikan kepada dua bahagian iaitu pelanggan dan pelayan. Tugas pelanggan ialah bertanggungjawab ke atas antara muka pengguna. Ia termasuk penyediaan antara muka pengguna bergrafik dan tindak balas ke atas input dari pengguna. Jika sistem pelanggan tidak berupaya bertindak balas terhadap permintaan pengguna, maka ia menyerahkan tugas tersebut kepada pelayan yang sesuai.

Pelayan pula melaksanakan tugas yang dikehendaki oleh pengguna. Tugas ini mungkin memerlukan sistem perniagaan, jabatan atau intranet, rangkaian nilai ditambah, atau LAN. Apa yang menarik ialah pelaksanaan pelayan tidak diketahui oleh pengguna. Setiap pengguna di sistem pelanggan hanya berinteraksi dengan antara muka pengguna bergrafik sahaja. Senibina ini dapat mengurangkan kos kerana kita boleh menggunakan stesen kerja atau komputer peribadi sebagai pelanggan dan pelayan.



Rajah 2.4 : Perhubungan Pelanggan – Pelayan

bersambung kepada rangkaian fizikalnya sendiri. TCP (Transmission Control Protocol) telah dibangunkan sebelum model OSI (iaitu model Open System Interconnection).

Ia merupakan suatu protokol dimana protokol bermaksud satu set peraturan dan prosedur untuk bertukar-tukar data di antara komputer dengan sebuah sistem rangkaian atau menerusi internet. Protokol biasanya meliputi maklumat yang melibatkan penyemakan ralat, pemampatan data, penghantaran dan penerimaan mesej. Protokol juga dikenali sebagai kaedah untuk mengakses. Protokol TCP/IP terdiri daripada 5 lapisan iaitu lapisan fizikal, penyambungan data, rangkaian, pengangkutan dan aplikasi.

Aplikasi (HTTP, FTP,....)
Pengangkutan (TCP, UDP)
Rangkaian (IP)
Penyambungan Data (Pemacu Peranti)
Fizikal (Penyambungan data)

Rajah 2.5 : Lapisan protokol TCP/IP

Secara umumnya, fungsi lapisan lapisan TCP/IP adalah seperti berikut :

Lapisan Aplikasi

Menyediakan komunikasi di antara proses – proses atau aplikasi – aplikasi pada hos yang berasingan.

2.2.2 Komponen Pelanggan – Pelayan

Pelanggan

Pelanggan boleh menghubungi komputer lain untuk mengguna atau mencapai sumber – sumber yang dikehendaki serta membuat permintaan. Komputer pelanggan biasanya terdiri daripada komputer peribadi atau stesen kerja yang bertindak sebagai bahagian depan (front-end) dalam satu rangkaian setempat dimana ia boleh digunakan dalam pertanyaan pangkalan data atau pencarian maklumat yang lain dari komputer pelayan.

Pelayan

Pelayan mempunyai perisian yang sentiasa beroperasi yang sentiasa menunggu permintaan yang akan dibuat oleh pelanggan serta menyediakan perkhidmatan pengurusan pangkalan data dan komunikasi. Komputer pelayan yang biasa digunakan mempunyai keupayaan pemprosesan yang lebih tinggi dan bertindak sebagai bahagian belakang (back-end) dan disambungkan kepada komputer pelanggan.

2.3 Komunikasi Komputer

Kebanyakan komputer yang digunakan di dalam rangkaian biasanya berkomunikasi dengan menggunakan *Transmission Control Protocol* (TCP) ataupun menggunakan *User Datagram Protocol* (UDP). Hos dalam TCP/IP adalah komputer. TCP/IP menganggap rangkaian fizikal yang bersambungan sebagai satu rangkaian yang besar. Ia menganggap semua hos disambungkan kepada rangkaian besar ini daripada

Lapisan Pengangkutan

Menyediakan perkhidmatan hujung-ke-hujung yang mungkin merangkumi mekanisma kebolehpercayaan. Ia menyembunyikan maklumat komprehensif rangkaian daripada aplikasi.

Lapisan Rangkaian

Bertanggungjawab ke atas penghalaan data daripada sumber ke hos destinasi melalui satu atau lebih rangkaian yang disambungkan oleh penghala (router).

Lapisan Penyambungan Data

Bertanggungjawab ke atas antaramuka logikal di antara sistem hujung (end-system) dan rangkaian.

Lapisan Fizikal

Mendefinisikan ciri – ciri medium transmisi, kadar data, isyarat dan skema pengkodan isyarat yang digunakan.

2.3.1 TCP / IP

Nama TCP/IP diambil dari singkatan *Transmission Control Protocol* dan *Internet Protocol* atau protokol kawalan transmisi dan protokol Internet. TCP/IP ialah nama biasa untuk sekumpulan lebih daripada 100 protokol yang digunakan bagi menyambungkan komputer dan rangkaian. Nama sebenar TCP/IP dibentuk dari dua protokol utama, iaitu TCP (Transport Control Protocol) dan IP (Internet Protocol).

Komputer yang membentuk rangkaian Internet 'bercakap' antara satu sama lain dengan menggunakan bahasa protokol TCP/IP. Protokol ini mengendalikan skema pengalamatan untuk setiap komputer dalam Internet. TCP/IP menentukan undang-undang dan peraturan bagaimana data seharusnya bergerak melalui rangkaian dan komputer. Dalam Internet, maklumat dihantar bukannya dengan cara gugusan data besar, tetapi sebaliknya ia dipecahkan kepada pakej-pakej kecil yang dipanggil paket. Pecahan data ini juga dikendalikan sebaiknya oleh TCP/IP.

Contohnya, katakan anda menghantar satu pesanan panjang kepada sahabat di negara lain, TCP akan membahagikan pesanan tersebut kepada sekumpulan paket. Setiap paket ditandakan dengan nombor turutan dan alamat penerima. Sebagai tambahan, TCP turut memasukkan maklumat kawalan ralat. Kemudian paket dihantar melalui jaringan. IP akan membuat penghantaran kepada destinasi yang dikehendaki. Di tempat destinasi tersebut, TCP menerima paket yang dihantar dan memeriksa sebarang ralat. Jika ada ralat, TCP meminta supaya paket tersebut dihantar semula.

Apabila semua paket diterima dalam keadaan baik, barulah TCP akan menggunakan turutan nombor bagi menjana semula pesanan seperti dalam keadaan asal. Dalam lain perkataan, tugas IP ialah mendapatkan data mentah, iaitu paket, dari satu tempat ke satu tempat. Manakala tugas TCP pula ialah memastikan aliran dan kandungan data yang dihantar adalah betul. Memecahkan data kepada paket mempunyai beberapa kebaikan utama.

Pertamanya, ia membenarkan ramai pengguna menggunakan talian komunikasi yang sama pada suatu masa. Memandangkan setiap paket tidak perlu dipindahkan bersama-sama, talian komunikasi boleh membawa pelbagai jenis paket yang menuju

destinasi yang berbeza. Bayangkan sebuah lebuh raya yang dipenuhi banyak kenderaan pada jalan yang sama, tetapi sebenarnya mereka menuju ke destinasi yang berasingan.

Dengan itu, TCP/IP merupakan sekumpulan protokol yang terlibat untuk mengendalikan peranti komputer dan komunikasi dalam rangkaian. Dengan adanya TCP/IP juga peranti komputer dan komunikasi boleh bekerjasama melaksanakan proses pemindahan data dari satu komputer hos ke komputer hos yang lain. Jawapan terbaik bagi TCP/IP ialah ianya merupakan 'gam' yang memegang beribu-ribu rangkaian dan berjuta-juta komputer dalam internet bersama-sama.

2.3.2 Protokol Internet (IP)

Pada lapisan rangkaian, protokol utama yang didefinisikan oleh TCP/IP adalah Protokol Internet (IP) walaupun terdapat protokol lain yang menyokong pengangkutan data dalam lapisan ini. IP merupakan satu protokol datagram yang tidak bersambung dan menggunakan perkhidmatan penghantaran usaha yang terbaik. Ia tidak menyediakan sebarang semakan ralat atau 'tracking'. IP mengangkut data dalam bentuk datagram, dimana setiap datagram akan diangkut secara berasingan.

Datagram merupakan bingkisan pelbagai lebar, yang terdiri daripada dua bahagian iaitu data dan juga header (mengandungi maklumat tambahan yang penting untuk penghalaan dan penghantaran). Setiap datagram mungkin diangkut melalui jalan (route) yang berlainan dan mungkin dalam susunan. IP tidak mengawal setiap laluan yang diambil oleh datagram tersebut dan tidak mempunyai kemudahan untuk menyusun semula datagram yang diterima. Oleh kerana ia adalah protokol yang tidak disambungkan, maka ia juga tidak mewujudkan litar maya untuk penghantaran mesej.

IP membekalkan fungsi transmisi yang membolehkan pengguna menambah kemudahan yang diperlukan untuk aplikasi yang tertentu dan membenarkan keberkesanan yang maksimum.

2.4 Perbandingan Sistem Sedia Ada

Terdapat banyak permainan komputer yang boleh didapati dari laman web di internet. Ini adalah disebabkan oleh percambahan pengguna yang lebih gemar bermain permainan komputer daripada permainan manual. Perbandingan dengan sistem sedia ada dibuat supaya pembangun sistem dapat mengenalpasti kebaikan serta kelemahan permainan sedia ada dengan permainan yang akan dibangunkan. Selain dari itu, ia juga dapat memberikan panduan dan memberi kefahaman yang lebih jelas terhadap permainan yang bakal dibangunkan.

2.4.1 Permainan Multi-Pengguna

Permainan ini khusus dibina untuk penggunaan multi-pengguna (lebih dari satu pemain). Pengguna dapat menikmati permainan ini bersama rakan – rakan ataupun dengan keluarga yang berada jauh dari pengguna. Dengan kewujudan permainan ini, pelbagai jenis permainan komputer adalah bercorak kerjasama antara dua pengguna atau lebih pada masa yang sama. Ia boleh dikategorikan kepada 3 jenis iaitu permainan aksi, permainan strategi dan juga permainan menguji minda.

Permainan Aksi (Action Game)

Pengguna boleh berinteraksi dengan keseluruhan persekitaran permainan tersebut dan pengguna juga dapat merasai seperti berada di dalam permainan itu sendiri. Ini adalah kerana grafik dan juga bunyi yang disediakan adalah seperti dalam keadaan nyata. Contoh permainan aksi adalah :

- Counter Strike

Counter Strike merupakan sebuah permainan berkonsep kumpulan, dimana salah satu kumpulan adalah pengganas dan pihak satu lagi adalah lawan daripada pengganas. Permainan ini menyediakan paparan antaramuka yang interaktif kerana senjata yang disediakan adalah seperti yang sebenar dan apabila pemain menembak sasaran, bunyi tembakan boleh didengari dengan jelas dan pengguna dapat merasa seperti benar – benar sedang menggunakan senjata tersebut.

Permainan Strategi (Strategy game)

Bagi permainan strategi, pengguna perlu memikirkan strategi bagaimana untuk memerangi dan memerangkap pihak lawan supaya dapat memenangi permainan tersebut. Contoh permainan strategi adalah :

- Starcraft

Ia adalah merupakan sebuah permainan yang dicipta oleh Blizzard, bagi pengguna yang gemar bermain permainan warcraft dan warcraft 2 atau yang pakar dalam strategi, mungkin pengguna tersebut akan menggemari permainan starcraft ini. Terdapat 3 makhluk luar angkasa yang hidup di

galaksi yaitu Terran, Protoss dan Zerg, ketiga – tiga makhluk tersebut akan bertempur sesama sendiri untuk mempertahankan diri masing – masing. Jadi masing – masing mempunyai strategi untuk saling menjatuhkan antara satu sama lain.

Permainan Menguji Minda (Brainsqueezer)

Permainan menguji minda boleh dimainkan oleh pelbagai lapisan pengguna, ianya berkonsep terbuka dan mudah untuk dimainkan di mana pengguna hanya perlu berfikir untuk memenangi permainan tersebut. Selain mematuhi peraturan dan juga langkah – langkah yang telah ditetapkan. Makin cergas pengguna berfikir, makin banyaklah peluang untuk pengguna tersebut memenangi permainan. Contoh permainan menguji minda ini adalah :

- Chess

Walaupun permainan ini boleh dimainkan secara manual tetapi pengguna lebih gemar bermain permainan ini di dalam komputer. Selain dapat bermain permainan, pengguna juga boleh berinteraksi dengan pengguna yang lain pada masa yang sama. Persekitaran antaramuka permainan yang interaktif menjadi aset penting dalam menarik minat pengguna dan sekaligus tidak membosankan pengguna di mana telah diketahui bahawa permainan chess mengambil jangkamasa yang agak lama untuk menamatkan permainan tersebut.

2.5.2 JavaScript

JavaScript merupakan bahasa skrip asas yang membenarkan pembangun laman Internet untuk membina halaman yang dapat berinteraksi dengan pengguna dengan menambahkan beberapa fungsi kecil kepada halaman. JavaScript adalah berasaskan sedikit daripada bahasa pengaturcaraan Java. JavaScript lebih fleksibel kepada pembangun laman web dengan menyediakan kemudahan untuk membina pelbagai aplikasi contohnya memaparkan teks bergerak, bunyi dan elemen multimedia yang lain dengan mudah.

Jadual 2.1 : Perbezaan antara Java dan JavaScript

Java	JavaScript
Bytecodes yang dikompil dari pelayan akan dilarikan pada pengguna.	Diterjemah (bukan dikompil) oleh pengguna.
Berorientasikan objek.	Berasaskan objek.
Applet diasingkan dari HTML.	Kod diterapkan dalam HTML.
Variable mesti diisytiharkan.	Variable tidak diisytiharkan.

2.6 Pendekatan Pengumpulan Maklumat Projek

Pengumpulan maklumat merupakan perkara yang amat penting sekali dalam membangunkan sesuatu sistem. Pembangunan sesuatu sistem tidak akan sempurna dan berjaya mencapai objektifnya jika tiada penyelidikan dan pengumpulan data mengenai sistem yang bakal dibangunkan. Terdapat pelbagai sumber yang membolehkan pembangun sistem memperoleh sebarang maklumat, sumber – sumber ini akan memberikan maklumat yang berbeza daripada penggunaan teknik yang berbeza.

Diantara sumber – sumber maklumat adalah pengguna sistem, borang kaji selidik, laporan, dokumen, aturcara komputer dan manual pengguna. Sumber maklumat yang utama adalah pengguna sistem. Ini adalah kerana pengguna sistem merupakan aset penting dalam menggunakan sesuatu sistem. Pengguna sistem yang akan menentukan bahawa sistem tersebut berkesan ataupun tidak. Melalui pengguna sistem juga, kita dapat mengetahui akan objektif, keperluan sistem serta aktiviti-aktiviti yang dijalankan pada sistem yang sedia ada.

Borang kaji selidik pula merupakan sumber di mana pembangun sistem dapat mengetahui kehendak dan keperluan pengguna. Ini adalah kerana sistem yang akan dibangunkan mestilah ramah pengguna dan juga memenuhi kehendak setiap pengguna. Selain itu, kita juga dapat memperoleh maklumat mengenai sesuatu aliran data sistem dan transaksi yang terlibat ketika proses pembangunan sistem dijalankan.

Laporan dapat digunakan dalam menentukan penjadualan projek dan juga aktiviti – aktiviti yang akan dijalankan. Ini adalah kerana laporan menyediakan maklumat output kepada pembangun sistem. Dokumen merupakan suatu bahan yang digunakan dalam mencari maklumat – maklumat yang berkaitan dengan sistem yang akan dibangunkan. Dalam membangunkan sesuatu sistem, kita tidak akan terlepas dengan penggunaan aturcara komputer. Terdapat pelbagai contoh aturcara komputer yang boleh digunakan oleh pembangun sistem. Contohnya, aturcara Java, C++, C, bahasa SQL dan pelbagai aturcara yang sesuai untuk sistem yang akan dibangunkan. Manual pengguna pula bertindak sebagai garispanduan dalam merekabentuk sistem yang terperinci. Pembangun sistem dapat menentukan rekabentuk antaramuka yang sesuai kepada pengguna sistem.

2.6.1 Kaedah Dan Teknik Dalam Pengumpulan Maklumat

Sebagaimana yang diketahui, terdapat pelbagai cara dalam mengumpulkan maklumat. Diantara kaedah dan juga teknik yang digunakan adalah :

Pengguna Sistem

Diantara kaedah yang boleh digunakan ketika mengumpulkan maklumat dari pengguna sistem adalah temubual secara tidak formal, iaitu dengan bertanya mengenai sistem yang pernah digunakan ataupun sistem yang mempunyai konsep yang sama dengan sistem yang bakal dibangunkan. Selain itu, pembangun sistem juga boleh menggunakan borang kaji selidik dan kemudian menganalisis borang kaji selidik tersebut.

Bilik Dokumen

Terdapat pelbagai maklumat yang boleh dijumpai di dalam bilik dokumen ini. Ia merupakan lokasi penyimpanan laporan – laporan projek terdahulu, buku – buku rujukan berkaitan komputer samada edisi lama ataupun edisi baru. Bahan – bahan yang terdapat didalam bilik dokumen ini banyak membantu dan memberikan garis panduan dalam membangunkan sesuatu sistem.

Internet

Nama asal bagi internet adalah ARPANET dimana ia merupakan rangkaian pertukaran packet yang digunakan untuk menyambungkan komputer kerangka utama empat universiti di United States pada lewat 1960 – an. Internet juga merujuk kepada sambungan dua atau lebih rangkaian.

Internet terdiri daripada pengguna dari organisasi mencari keuntungan dan juga yang bukan mencari keuntungan. Kegunaan internet semakin bertambah dan digunakan untuk menghantarkan fail, menukar mesej dan mencari maklumat. Maklumat – maklumat yang dikehendaki boleh dicari dengan menggunakan enjin pencarian. Enjin pencarian adalah suatu perkakasan pencari yang dapat mencari maklumat dalam World Wide Web (WWW) berdasarkan kata kunci yang ditekankan.

Diantara contoh enjin pencarian yang terdapat di internet :

- www.yahoo.com
- www.google.com
- www.altavista.com
- www.whatis.com
- www.lycos.com

Perpustakaan

Maklumat – maklumat yang berkaitan dengan sistem yang bakal dibangunkan boleh diperolehi dari perpustakaan. Perpustakaan menyediakan pelbagai buku samada yang berkaitan dengan komputer ataupun tidak.

Bab 3: Metodologi

University of Malaysia

Bab 3 : Metodologi

Sebelum memulakan pembangunan sistem, perancangan terhadap sistem tersebut perlulah dilaksanakan terlebih dahulu. Metodologi pembangunan menyediakan pelbagai jenis model – model pembangunan di mana ia boleh digunakan sebagai panduan bagi melaksanakan fasa – fasa yang berikutnya. Diantara contoh model pembangunan adalah model air terjun, model prototaip, model pembangunan berfasa, model spiral, model V dan lain – lain.

3.1 Model Air Terjun

Metodologi pembangunan yang digunakan adalah Model Air Terjun. Model ini merupakan satu kaedah klasik dan diamalkan meluas dalam pembangunan sistem. Ia adalah satu siri aktiviti berjujukan dan bersistematik. Selain daripada digunakan dengan meluas, ia juga memudahkan aktiviti pengurusan projek kerana masa untuk mula dan akhir sesuatu fasa ditentukan dengan jelas. Kalau hendak dibandingkan model air terjun dengan model pembangunan yang lain, model air terjun adalah lebih teguh dan mudah untuk diubahsuai.

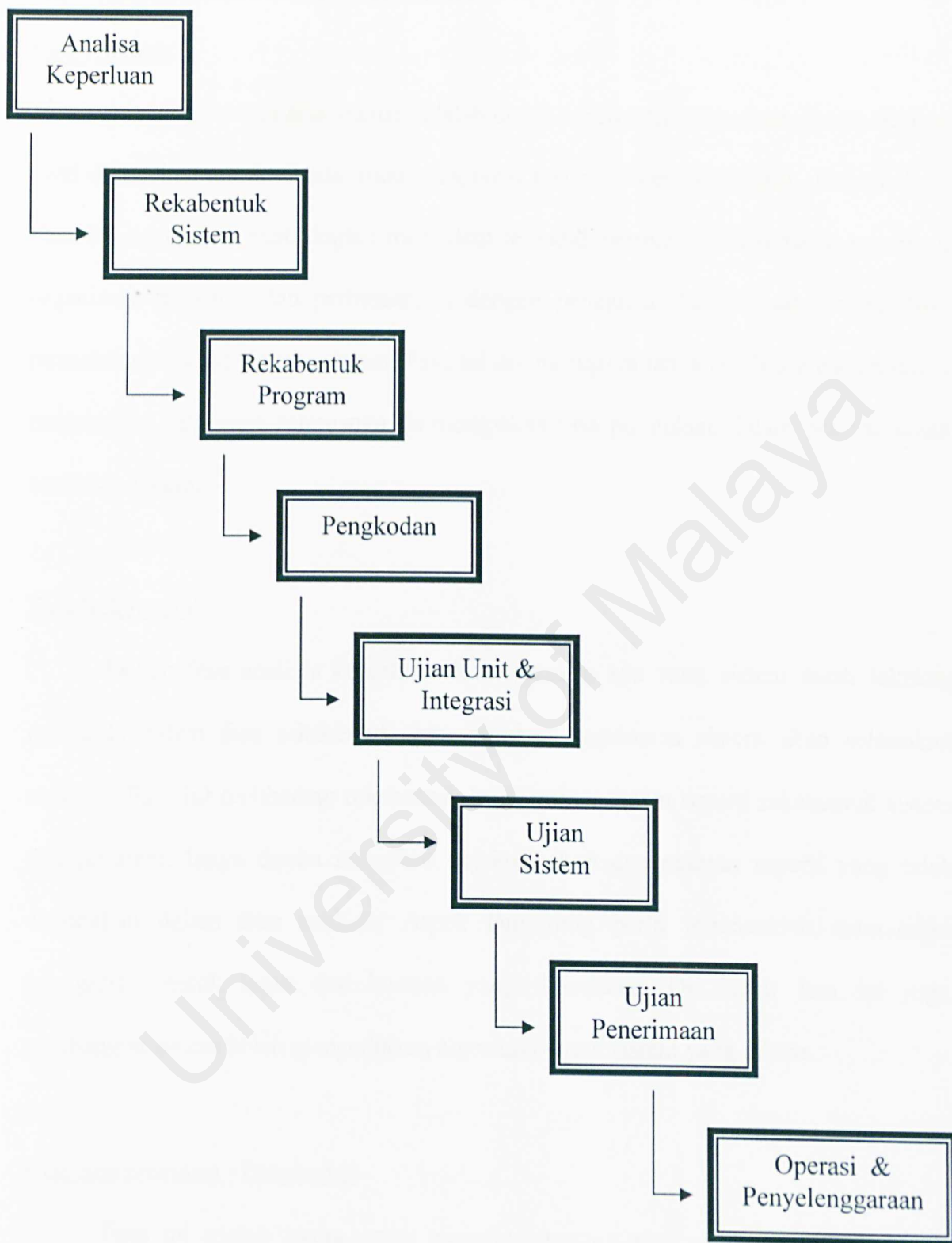
Fasa –fasa yang terdapat dalam model ini adalah fasa – fasa utama yang terdapat dalam membangunkan sesuatu sistem. Fasa – fasa yang terlibat adalah fasa analisis, fasa rekabentuk, fasa implementasi / pengkodan, fasa pengujian dan juga fasa penyelenggaraan. Aktiviti – aktiviti pembangunan sistem berlaku secara linear daripada satu fasa ke fasa yang berikutnya.

Dengan lain perkataan, aktiviti – aktiviti dalam fasa rekabentuk hanya boleh dilakukan selepas semua aktiviti di dalam perancangan sistem disiapkan dengan sempurna. Aktiviti dalam fasa implementasi / pengkodan tidak boleh dilakukan selagi aktiviti fasa rekabentuk belum selesai, dan begitulah seterusnya.

Hasil kerja sesuatu fasa akan mempengaruhi fasa – fasa berikutnya, oleh itu ia perlu disemak dan dinilai sebelum melangkah ke fasa yang seterusnya. Setiap fasa akan menghasilkan keluaran dimana ia boleh digunakan untuk fasa yang berikutnya.

Model Air Terjun dipilih dalam membangunkan sistem ini kerana :

- Model ini mudah diterangkan kepada pengguna yang tidak biasa dengan pembangunan sistem.
 - Ianya berupaya mencadangkan jujukan yang perlu dibangunkan terlebih dahulu.
 - Pembangun sistem dapat melihat apa yang berlaku pada setiap fasa dengan mudah.
 - Pembangun sistem dapat mengenalpasti keperluan – keperluan yang diperlukan dalam sistem yang akan dibangunkan dan ianya bersesuaian dengan kehendak pengguna.
 - Pembangunan sistem yang teratur kerana ianya berperingkat – peringkat.
- Menumpukan sepenuh perhatian pada setiap fasa yang terlibat.



Rajah 3.1 : Model Air Terjun

3.2 Fasa Umum Pembangunan Sistem

Fasa Analisis

Matlamat utama fasa analisis adalah untuk menyelidiki keperluan sistem. Kajian awal diperlukan untuk mendapatkan maklumat tentang sistem yang akan dibangunkan. Fasa ini merupakan suatu kajian mendalam terhadap permasalahan semasa, keperluan organisasi, piawaian dan perbincangan dengan pengguna. Ianya amat penting bagi memahami objektif sistem sebenar. Fasa ini dibina supaya ianya boleh digunakan untuk meneruskan fasa yang seterusnya. Ia merupakan fasa permulaan dalam pembangunan sesebuah sistem.

Fasa Rekabentuk

Dalam fasa analisis keperluan, kita tentukan apa yang sistem mesti lakukan manakala dalam fasa rekabentuk, kita jelaskan bagaimana sistem akan selesaikan masalah. Fasa ini melibatkan rekabentuk keseluruhan sistem seperti rekabentuk sistem dan program. Ianya direka mengikut objektif, skop dan sasaran seperti yang telah dinyatakan dalam fasa analisis. Aspek lain yang perlu termasuklah antaramuka pengguna, bentuk input dan laporan yang dihasilkan. Di dalam fasa ini juga, pembangun sistem boleh menyediakan algoritma untuk sistem yang dibina.

Fasa Implementasi / Pengkodan

Fasa ini adalah usaha untuk menukar aturcara atau menterjemah rekabentuk terperinci dalam fasa rekabentuk kepada kod aturcara. Ia juga dikenali sebagai penulisan aturcara dan dilakukan oleh pengaturcara. Jika ada ralat semasa

pengkompilan, aturcara diperbetul dan dikompil sehingga tiada ralat lagi. Fasa ini menjadi lebih mudah jika fasa analisis dan fasa rekabentuk dilakukan dengan betul dan lengkap.

Fasa Pengujian

Sesuatu sistem perlu disemak dan disahkan sebelum diserahkan kepada pengguna. Fasa pengujian dilakukan apabila kesemua fasa – fasa sebelumnya telah siap dan semua unit telah digabungkan. Pengujian sistem yang dilakukan akan membolehkan pembangun sistem mengenalpasti ralat, menentukan keupayaan sistem dan keserasian sistem yang dibangunkan dengan keperluan pengguna. Selain dari itu, ia juga dapat mengukur dan menilai kualiti serta kebolehpercayaan sistem yang akan dihasilkan.

Fasa Penyelenggaraan

Setelah aturcara berjaya diuji, bermakna ia boleh beroperasi seperti yang dikehendaki. Fasa ini merupakan peringkat pertama penggunaan sistem, di mana sistem akan dipasang dan pengguna boleh menggunakan sistem tersebut. Fasa penyelenggaraan dibuat apabila ada keperluan untuk perubahan dalam sistem yang sedang beroperasi. Biasanya kesilapan atau ralat yang dikesan tidak berapa serius dan perubahan akan dibuat pada penghujung setiap fasa sekiranya ia diperlukan. Fasa ini dilakukan sekiranya ralat tidak dapat dikesan semasa pengujian, keperluan pengguna baru, persekitaran peralatan yang berubah dan juga menambahkan fungsi baru pada sistem.

3.3 Model Pembangunan Yang Lain

Selain daripada model air terjun, saya ada mendapatkan sedikit maklumat mengenai beberapa lagi model pembangunan yang lain sebagai perbandingan dan pengetahuan umum sahaja.

3.3.1 Model Prototaip

Dengan menggunakan model prototaip, keperluan pengguna yang diperolehi dari awal pembangunan sistem memang diakui tidak lengkap. Berdasarkan maklumat awal tentang keperluan pengguna ini, rangka sistem akan dibina. Prototaip mungkin dibina untuk mewakili keseluruhan fungsian mengikut keperluan awal pengguna. Ia juga mungkin dibina hanya untuk mewakili bahagian – bahagian yang kritikal dan sukar difahami. Prototaip yang telah dibangunkan oleh pembangun sistem akan dilaksanakan oleh pengguna untuk disahkan.

Terdapat dua pendekatan di dalam model prototaip:

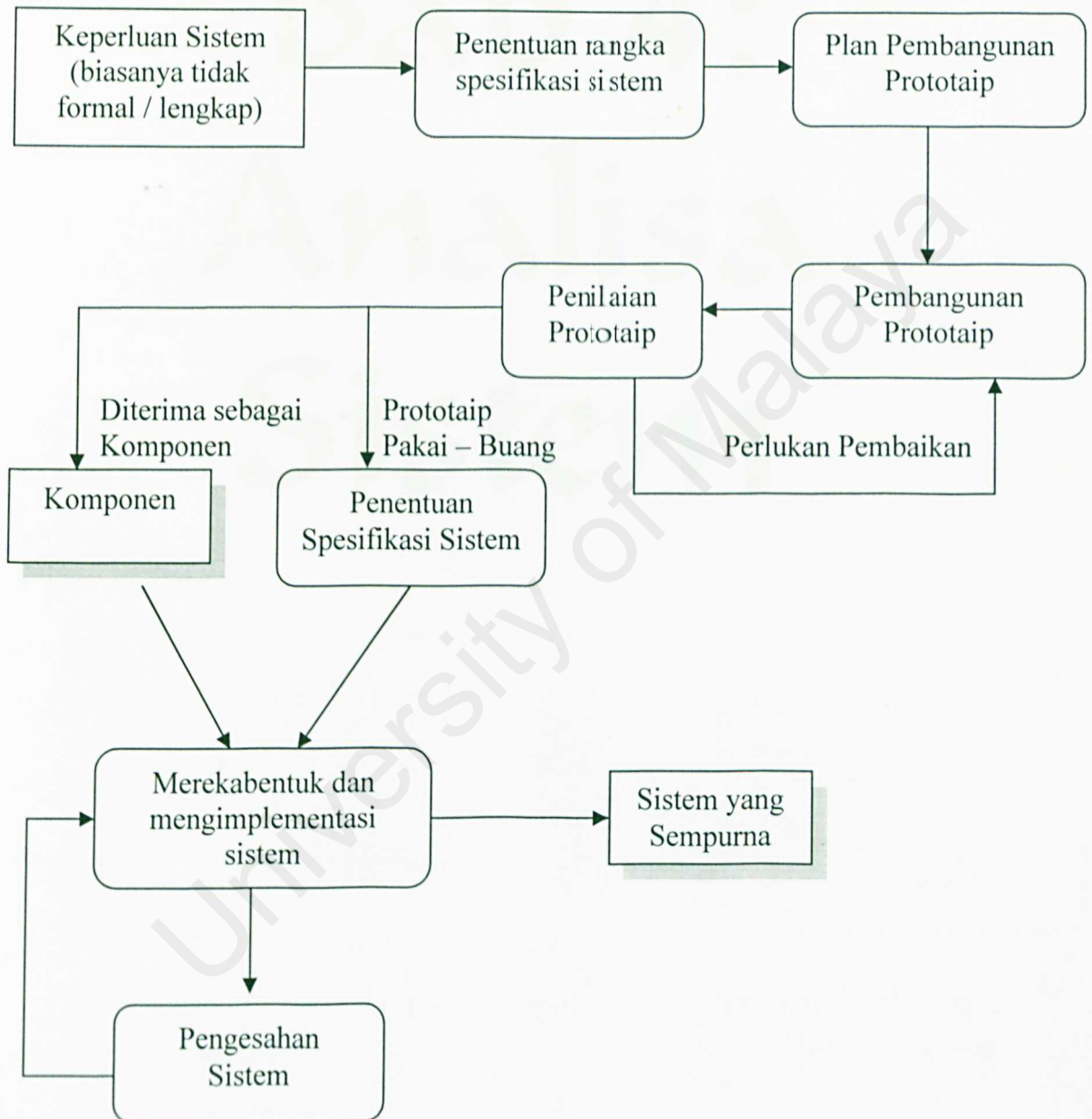
a) Prototaip penjelajahan

Di dalam prototaip penjelajahan, prototaip yang disahkan akan digunakan sebagai komponen kepada sistem yang akan dibangunkan. Beberapa perubahan kecil kepada produk teras seterusnya akan dilakukan di peringkat pembangunan sebenar sistem. Sistem ini kemudiannya akan diuji dan disahkan sebelum dihantar kepada pengguna.

b) Prototaip pakai – buang

Bagi kes prototaip yang tersasar jauh daripada kehendak sebenar pengguna, masa dan kos untuk memperbaiki prototaip adalah tinggi. Untuk kes ini,

prototaip yang dibina akan dibuang atau dibaikan. Apabila prototaip dibuang, ia tidak bermakna pembinaan prototaip adalah membazir masa dan tenaga. Apa yang diperolehi disini adalah keperluan sebenar pengguna telah diperolehi.



Rajah 3.2 : Model Prototaip

Bab 4: Analisa Sistem

Bab 4 : Analisa Sistem

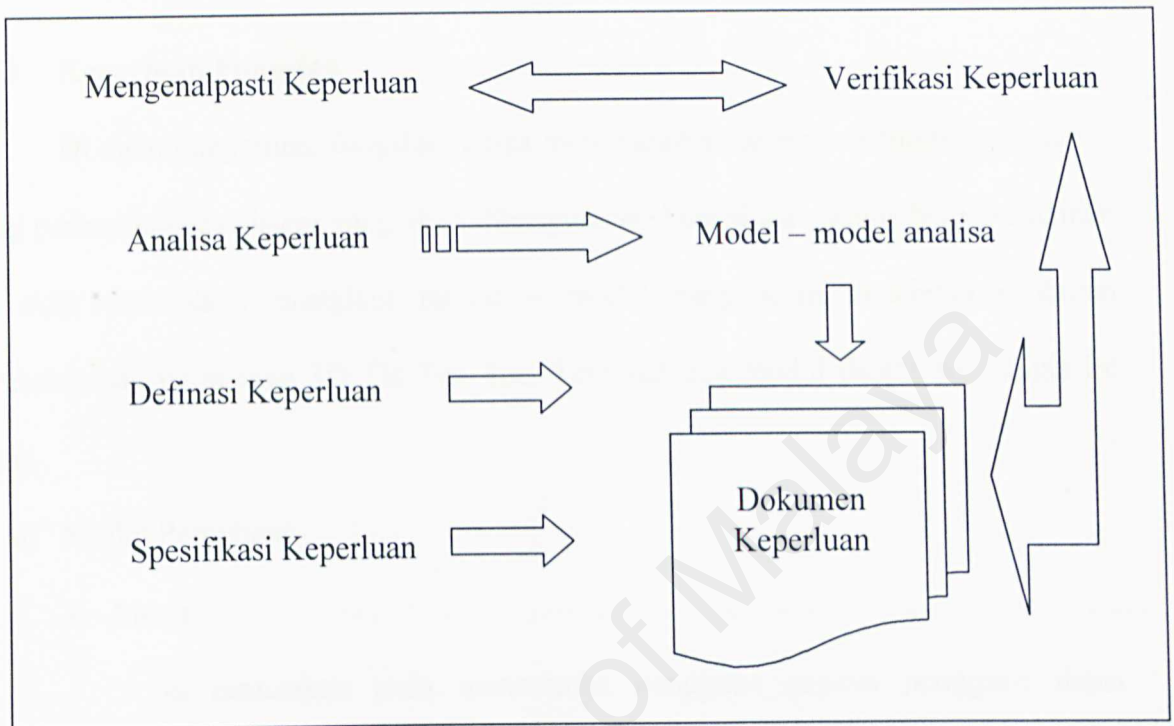
Analisa Sistem dilaksanakan bagi menentukan analisa keperluan yang diperlukan dalam membangunkan sebuah sistem. Analisa keperluan diperolehi hasil analisa dari pelbagai sumber. Ianya perlu untuk mendapatkan definisi keperluan yang tepat. Analisa yang dilaksanakan mestilah mencukupi dalam konteks terdapatnya tahap yang boleh diterima bagi risiko yang berkaitan dengan teknikal dan ketersauran kos, di samping tahap yang berkaitan dengan kesempurnaan, ketepatan dan kurang kesamaran dalam sistem yang akan dibangunkan.

Mengenalpasti keperluan adalah merupakan langkah pertama dalam kejuruteraan keperluan. Ia meliputi keperluan dari pengguna ataupun diperolehi daripada keperluan sistem. Keperluan adalah pernyataan – pernyataan yang menghuraikan sistem yang hendak dibangunkan dalam semua aspek secara jelas dan selari. Kejuruteraan keperluan adalah proses mengenalpasti servis (fungsi yang mesti ada pada sistem) dan kekangan sistem yang hendak dibangunkan. Kejuruteraan keperluan boleh didefinisikan sebagai proses mengenalpasti, menganalisa dan memodelkan keperluan perisian.

Empat aktiviti utama dalam kejuruteraan keperluan iaitu :

- Mengenalpasti keperluan – mendapatkan maklumat keperluan sistem dan memahami kehendak pengguna terhadap sistem yang hendak dibangunkan.
- Analisa keperluan – menganalisa setiap keperluan yang telah dikumpulkan. Hasil analisa akan diterjemahkan dalam bentuk model.
- Definisi dan spesifikasi keperluan – merekodkan keperluan sistem dengan jelas dan terperinci dalam bentuk dokumen.

- Verifikasi keperluan – memastikan spesifikasi keperluan adalah sama dengan keperluan sistem yang sebenar, mematuhi piawaian dan boleh digunakan sebagai asas bagi rekabentuk peringkat awal.



Rajah 4.1 : Proses Kejuruteraan Keperluan

Mengenalpasti keperluan adalah merupakan langkah pertama dalam kejuruteraan keperluan. Ia meliputi aktiviti mendapatkan keperluan daripada pengguna ataupun diperolehi dari keperluan sistem.

Keperluan boleh dibahagikan kepada dua kategori iaitu keperluan sistem dan juga keperluan alatan pembangunan sistem. Keperluan sistem dipecahkan kepada dua iaitu *keperluan fungsian* dan *keperluan bukan fungsian* manakala keperluan alatan pembangunan diterangkan mengenai *keperluan perisian* dan *keperluan perkakasan*.

4.1 Keperluan Sistem

Keperluan sistem boleh dikelaskan kepada dua kategori iaitu keperluan fungsian dan keperluan bukan fungsian.

4.1.1 Keperluan Fungsian

Di dalam keperluan fungsian, ianya menghuraikan mengenai fungsi atau servis yang perlu ada pada sistem yang akan dibangunkan. Fungsi atau servis bagi permainan ini akan diwakilkan mengikut modul – modul yang telah dicadangkan dalam pembangunan permainan 3D Tic Tac Toe. Terdapat tiga modul dalam permainan ini iaitu:

a) Modul Permainan

➤ Menu

Ia dimuatkan pada antaramuka pengguna supaya pengguna dapat memperolehi arahan daripada pembangun sistem dengan jelas. Menu yang dipaparkan adalah merupakan teks dan juga butang yang mana ia menerangkan setiap arahan yang diperlukan oleh pengguna contohnya pemilihan samada ingin meneruskan permainan ataupun tidak. Jika memilih untuk meneruskan permainan maka kotak permainan akan dikosongkan.

Selain itu, terdapat juga penerangan tentang keputusan permainan – siapa pemenang, permainan dimulakan atau ditamatkan dan juga maklumat-maklumat lain. Peringatan untuk menunggu giliran bermain bagi permainan rangkaian (sesama pengguna).

➤ Peletakan elemen- elemen

Elemen - elemen yang digunakan adalah dalam bentuk grafik 2 dimensi. Ianya diletakkan pada kedudukan yang bersesuaian dan tidak mengganggu aplikasi yang lain. Contoh elemen yang digunakan seperti papan TIC TAC TOE, nod bulat dan pangkah, butang atau ikon dan juga objek - objek tertentu yang terdapat dalam permainan.

b) Modul Pelanggan – Pelayan

➤ Pelanggan

Dihubungkan kepada pelayan. Ia bertindak menghantar maklumat seperti nod yang digunakan dan juga menghantar segala tindakan yang dilakukan oleh pengguna semasa permainan diaktifkan. Memaparkan status permainan.

➤ Pelayan

Menerima maklumat nod dan perletakan nod kemudian mengumpulkan keseluruhan data mengenai pemain kemudian mengaktifkan permainan. Menghantar maklumat status terkini kepada pengguna mengikut giliran setiap pemain. Sentiasa berfungsi bagi mengawal status permainan.

a) Modul Kotak Perbualan

➤ Kata- kata

Beberapa pilihan kata - kata disediakan terlebih dahulu dan bagi pengguna yang ingin menghantar mesej, hanya kata- kata yang telah disediakan

sahaja boleh dihantar kepada pengguna lain. Contohnya “You Win”, “You Lose” dan sebagainya. Ia adalah bagi mengelakkan daripada perbualan yang tidak diingini berlaku dalam kotak perbualan tersebut.

4.1.2 Keperluan Bukan Fungsian.

Bagi keperluan bukan fungsian pula, ia menyatakan mengenai kekangan ke atas sistem yang menyebabkan pilihan kita dalam membangunkan penyelesaian masalah dihadkan. Ianya bagi menambahkan kefahaman terhadap sistem yang bakal dibangunkan. Antara keperluan bukan fungsian yang perlu dipertimbangkan adalah :

- ❖ **Antaramuka yang mesra pengguna**

Pembangunan antaramuka pengguna mestilah dilakukan dengan teliti kerana ianya merupakan perkara pertama pengguna berinteraksi. Ianya penting untuk memastikan pengguna memahami bagaimana untuk menggunakan sistem dengan sekali pandang sahaja. Selain daripada itu, ia juga bagi menarik minat pengguna terhadap permainan tersebut.

- ❖ **Masa Tindak Balas**

Untuk memastikan sebuah sistem itu berjalan dengan lancar, masa yang diperlukan oleh sistem untuk memberi malumbalas setelah sesuatu input dimasukkan adalah dipastikan kurang daripada 30 saat. Bagi memastikan ianya menepati masa tersebut maka imej - imej yang digunakan adalah tidak terlalu banyak dan bersaiz kecil.

❖ Kebolehpercayaan

Projek permainan yang akan dibangunkan perlulah stabil dalam melaksanakan tindakan - tindakan yang dikehendaki oleh pengguna. Ini bermakna maklumat yang disampaikan adalah berwibawa, mempunyai sifat keaslian dan berautoriti.

❖ Senang Dikendalikan

Perhubungan antara setiap butang yang diletakkan pada permainan mestilah dapat memudahkan pengguna melaksanakan setiap arahan permainan tersebut. Contohnya, penggunaan ikon yang berpadanan dengan fungsi sistem, penggunaan warna dan cara memasukkan sesuatu arahan juga hendaklah konsisten.

❖ Ketepatan

Keputusan yang dibuat berdasarkan maklumat yang tidak tepat merupakan keputusan yang buruk. Ketepatan merujuk kepada sejauh mana maklumat itu bebas daripada kesalahan. Segala keluaran yang dipaparkan mestilah tepat dan sentiasa diperbaharui supaya pengguna tidak keliru.

4.2 **Keperluan Alatan Pembangunan**

Keperluan alatan pembangunan adalah berdasarkan kepada teknologi - teknologi yang wujud dan sesuai untuk digunakan dalam proses pembangunan sistem. Di dalam keperluan alatan terdapat dua jenis keperluan yang perlu diambil kira.

4.2.1 **Keperluan Perkakasan**

Perkakasan adalah penting untuk membangunkan sebuah sistem. Keperluan perkakasan boleh dikelaskan kepada dua bahagian iaitu perkakasan yang perlu ada pada pembangun sistem dan juga perkakasan yang perlu ada pada pengguna bagi melaksanakan sistem tersebut.

Jadual 4.1: Keperluan Perkakasan Pembangun Sistem dan Pengguna

Sistem Operasian (OS)	Microsoft Windows 2000
Pemproses (CPU)	450 MHz atau lebih
Ingatan Capaian Rawak (RAM)	Minimum 64 MB
Cakera Keras (Hard Disk)	10 GB atau lebih
Kad Grafik	3D Grafik – AGP 2X
Tetikus	PS2
CD – ROM	36 X Min
Cakera Liut (Disket)	1.44 MB

4.2.2 Keperluan Perisian

Bagi keperluan perisian, ianya menerangkan tentang perisian - perisian yang digunakan dalam merekabentuk imej yang terdapat pada antaramuka pengguna dan juga untuk melaksanakan aplikasi Java. Ia boleh dikelaskan kepada dua iaitu perisian yang digunakan oleh pembangun dan juga perisian yang diperlukan oleh pengguna.

Untuk melaksanakan aplikasi Java, kita memerlukan Kit Pembangunan Java versi 1.3 (JDK 1.3) di mana ia digunakan untuk memastikan kelajuannya adalah sama seperti bahasa pengaturcaraan C++. JDK 1.3 mengandungi perisian dan peralatan di mana pembangun sistem memerlukannya untuk mengkompil, menyahpijat dan melarikan applet serta aplikasi yang ditulis dalam bahasa Java.

Jadual 4.2: Keperluan Perisian Pembangun Sistem

Bahasa Pengaturcaraan Pelayan	Java
Bahasa Pengaturcaraan Pelanggan	Java
Pelaksanaan Pengaturcaraan	Java Development Kit (JDK 1.3)
Pelayar Web	Internet Explorer 4.0 atau lebih, Netscape Communicator 4.5 atau lebih
Aturcara Halaman Web	Java
Rekebentuk Antaramuka Pengguna	JCreator, Notepad, J++
Rekabentuk Grafik	Adobe Photoshop, Paint
Dokumentasi dan Penjadualan Projek	Microsoft Word 2000, Microsoft Powerpoint 2000
Lain- lain	Norton AntiVirus 2001

Jadual 4.3: Keperluan Perisian Pengguna

Bahasa Pengaturcaraan Pelanggan	Java
Pelaksanaan Pengaturcaraan	Java Development Kit (JDK 1.3)
Paparan Kod Program	JCreator, Notepad, J++
Pelayar Web	Internet Explorer 4.0 atau lebih, Netscape Communicator 4.5 atau lebih
Lain- lain	Norton AntiVirus 2001

4.2.3 Keperluan alatan lain

Pencetak dan pengimbas untuk membantu dalam pembangunan sistem secara am. Alatan- alatan ini digunakan untuk mencetak dan mengimbas dokumen - dokumen serta gambar- gambar yang perlu digunakan dalam pembangunan sistem. Selain itu juga, Pemacu Cakera Padat Baca dan Rekod (CD-RW) untuk merekod segala data – data yang berkaitan dengan sistem yang dibangunkan.

Bab 5: Rekabentuk Sistem

Bab 5 : Rekabentuk Sistem

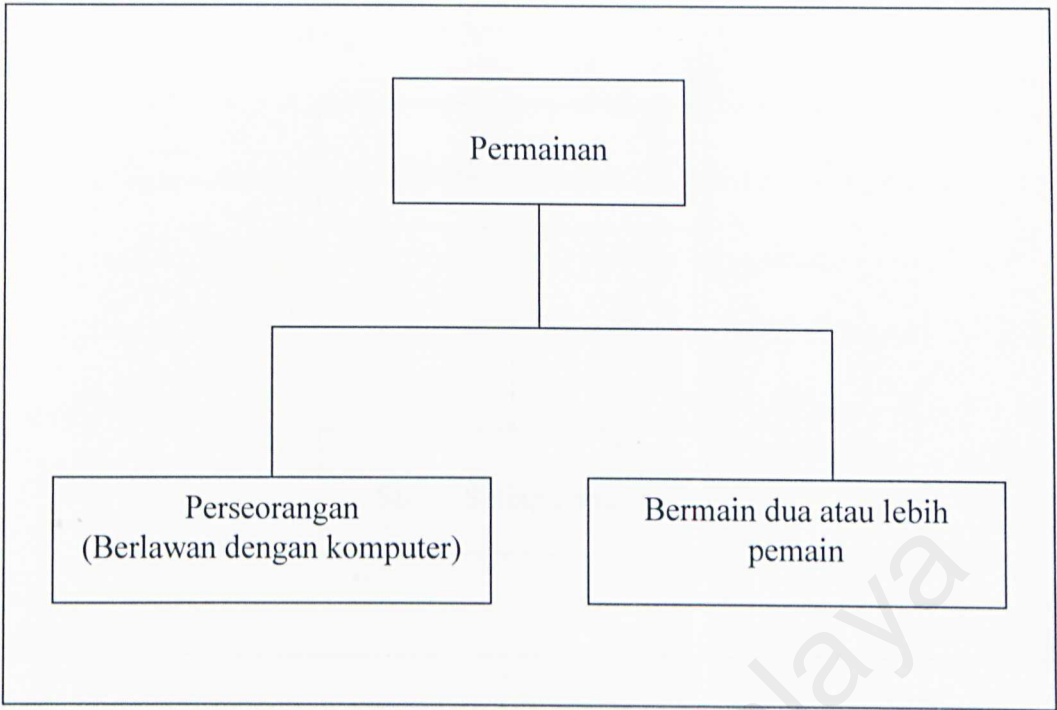
Rekabentuk sistem merangkumi semua tugas dan fungsi yang memberi keutamaan kepada spesifikasi terperinci dan mendalam berasaskan kepada penyelesaian masalah berdasarkan komputer. Hasil daripada proses analisa keperluan perlu dipersembahkan dalam bentuk yang mudah difahami. Terdapat dua jenis pendekatan yang boleh digunakan bagi mempersembahkan hasil analisa. Pendekatan analisa berasaskan kaedah adalah lebih sistematik bagi memodelkan sistem. Secara amnya, sistem model adalah berasaskan konsep pengaturcaraan, iaitu objek dan fungsi.

5.1 Rekabentuk Berorientasikan Fungsian

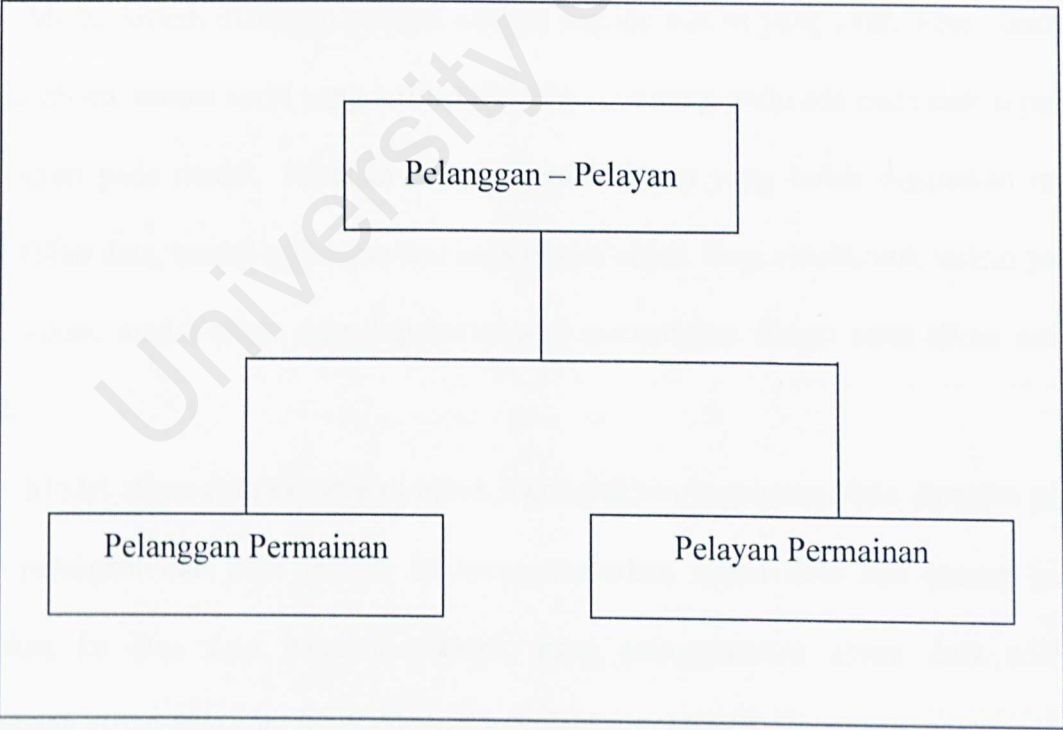
Ianya merupakan pendekatan pembangunan sistem kepada implementasi keperluan – keperluan yang telah dinyatakan. Rekabentuk ini akan menterjemahkan keperluan fungsian sistem kepada fungsian sistem. Pendekatan yang digunakan dalam rekabentuk ini adalah pendekatan atas – bawah di mana modul aras tinggi menerangkan sistem umum manakala modul aras bawah adalah lebih spesifik.

5.1.1 Carta Struktur

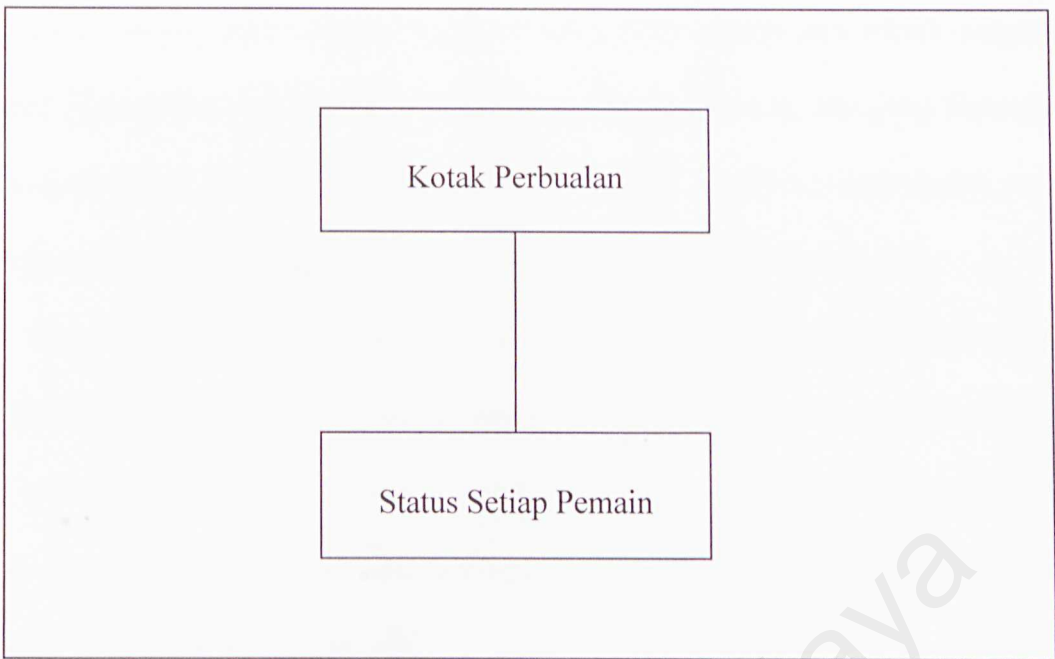
Carta struktur merupakan perwakilan grafik berdasarkan modul – modul fungsian sistem. Rajah - rajah seterusnya menunjukkan carta struktur untuk modul permainan 3D Tic Tac Toe, modul pelanggan – pelayan dan modul kotak bualan. Ia menerangkan secara ringkas apa yang terlibat dalam setiap modul.



Rajah 5.1 : Modul Permainan



Rajah 5.2 : Modul Pelanggan – Pelayan



Rajah 5.3 : Modul Kotak Perbualan

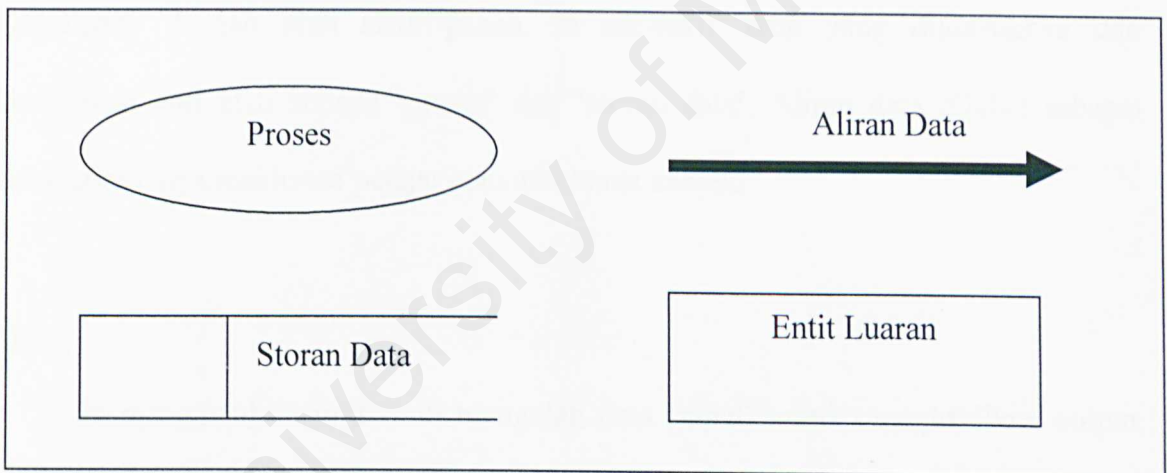
5.1.2 Model Aliran Data

Model sistem dianggap sebagai abstrak kepada sistem yang akan dibangunkan. Oleh sebab itu, semua entiti yang terlibat dan ciri - ciri yang perlu ada pada sistem perlu ditunjukkan pada model. Terdapat 3 jenis model sistem yang boleh digunakan iaitu model aliran data, model gabungan dan juga model objek. Bagi rekabentuk sistem yang dibangunkan, model aliran data digunakan bagi menentukan fungsi serta aliran setiap modul.

Model aliran data digunakan untuk menunjukkan bagaimana data diproses pada setiap peringkat dan juga modul. Ia menggambarkan aliran data dan proses yang dilakukan ke atas data tersebut. Teknik yang menggunakan aliran data adalah gambarajah aliran data atau lebih dikenali sebagai DFD (data flow diagram).

DFD juga dikenali sebagai '*bubble chart*'. DFD adalah satu teknik bergrafik yang menggambarkan aliran maklumat dan juga perubahan bentuk data yang digunakan sebagai pergerakan data daripada input kepada output. Teknik ini bersesuaian untuk sistem pemprosesan data kerana ia berorientasikan kaedah analisis berstruktur.

Pergerakan data dan proses – proses yang terlibat dapat ditunjukkan dengan jelas melalui teknik ini. Setiap proses merupakan aktiviti yang akan diproses input untuk menghasilkan beberapa output. Output daripada satu proses akan digunakan sebagai input kepada proses yang berikutnya. Terdapat empat simbol asas yang digunakan sebagai notasi- notasi DFD.



Rajah 5.4 : Notasi – notasi Gambarajah Aliran Data (DFD)

Pada dasarnya penggunaan beberapa jenis teknik DFD ini adalah sama, terdapat sedikit kelainan dari beberapa aspek, contohnya bentuk simbol yang digunakan. Walaubagaimanpun, kita boleh merumuskan teknik- teknik tersebut sebagai DFD.

DFD boleh dipecahkan kepada beberapa paras DFD untuk menggambarkan aliran data dan fungsi dengan lebih terperinci.

Kelebihan model aliran data adalah seperti berikut:

- ❖ Membantu dari segi kefahaman terhadap sistem yang akan dibangunkan.
- ❖ Mudah dibangunkan dan juga melakukan penyemakan.
- ❖ Membenarkan beberapa aras gambaran.

Penerangan tentang fungsi setiap notasi DFD:

Aliran Data

Laluan data bergerak dari satu bahagian ke bahagian lain. Arah laluan dikenalpasti dengan arah anak panah. Ia mewakili data yang dimasukkan dan dikeluarkan dari atau kepada 'proses' dan 'storan data'. Aliran data dilabel sebagai nama, contohnya maklumat pelajar atau maklumat subjek.

Proses

Ia menggambarkan aktiviti mengolah data (input) untuk menghasilkan output. Setiap proses mesti mempunyai sekurang- kurangnya satu input dan satu output. Nombor yang terdapat pada proses mestilah unik kerana ianya adalah nombor rujukan proses. Proses dilabelkan sebagai perbuatan, contohnya mengira markah, mendapatkan gred, mencetak slip dan sebagainya.

Entiti Luaran

Sumber luaran yang berinteraksi dengan sistem yang akan dibangun. Entiti luaran menyediakan data atau sumber kepada sistem atau menerima maklumat dari sistem. Walau bagaimanapun, komunikasi di antara entiti luaran adalah di luar dari skop sistem yang dikaji. Ia dilabel sebagai kata nama, contohnya pelajar, pensyarah, pentadbir.

Storan Data

Digunakan untuk menyimpan data bagi kegunaan proses. Lazimnya, data - data untuk kegunaan proses akan disimpan dalam pangkalan data, pita magnetik atau mikrofilem secara fizikal. Walau bagaimanapun, DFD tidak mempertimbangkan tentang bagaimana data - data tersebut disimpan secara logikalnya. Storan data dilabel sebagai kata nama, contohnya rekod pelajar, rekod matapelajaran.

5.2 Rekabentuk Sistem Yang Dibangunkan

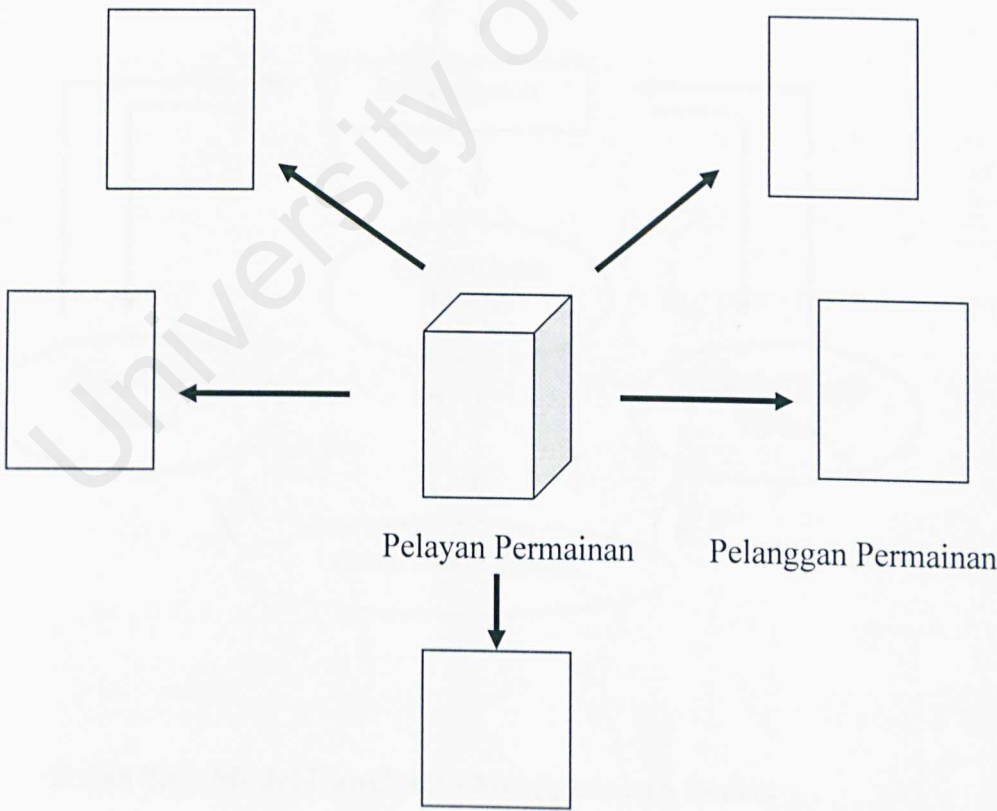
Di dalam rekabentuk sistem ini boleh dibahagikan kepada 2 pecahan iaitu rekabentuk pelanggan - pelayan dan rekabentuk permainan.

5.2.1 Rekabentuk pelanggan - pelayan

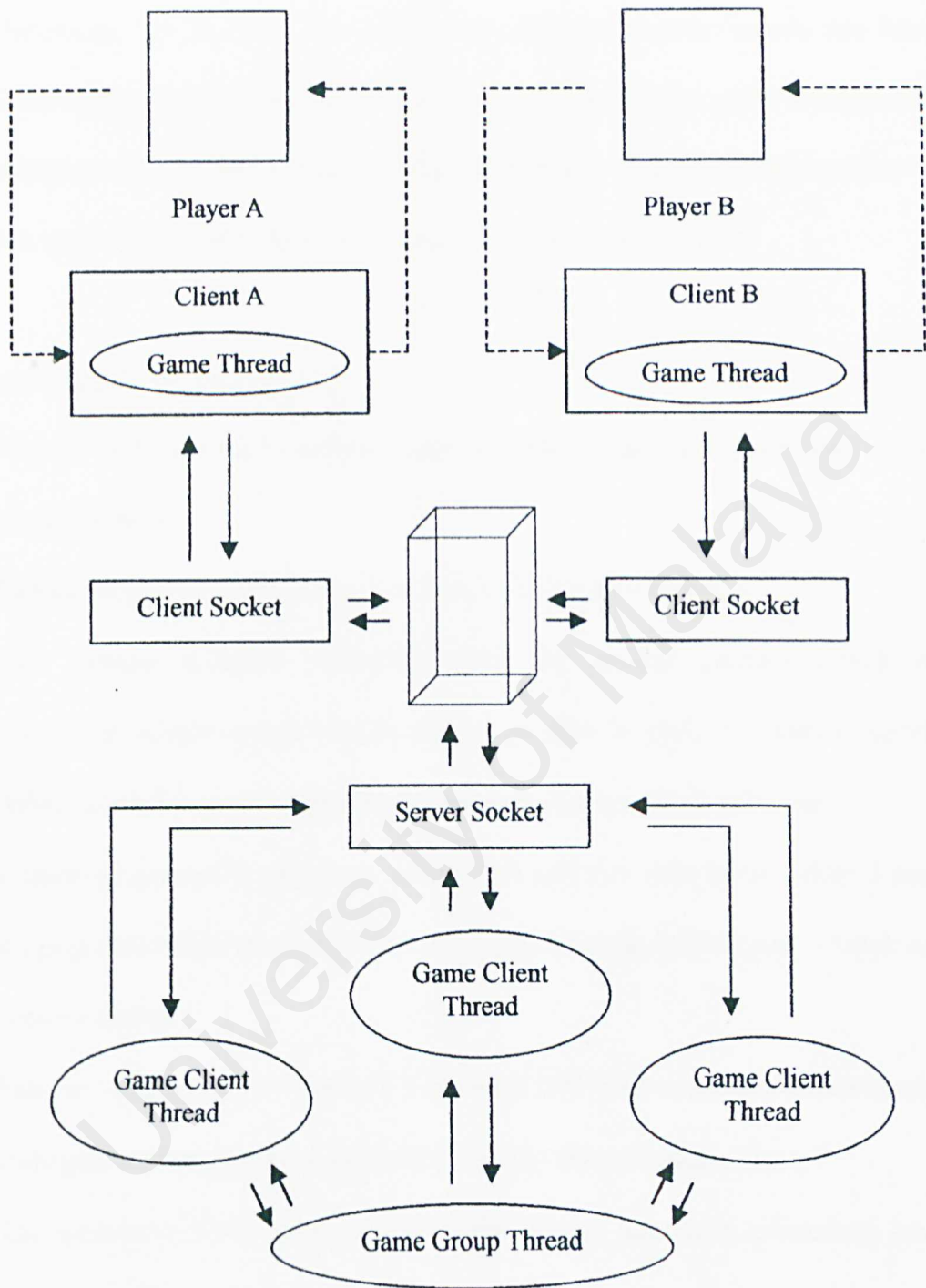
Model umum dari pengaturcaraan rangkaian adalah pengaturcaraan pelanggan – pelayan di mana prinsipnya adalah sederhana. Pelanggan akan menghantar signal permintaan ke pelayan untuk minta data atau infomasi yang disediakan di pelayan. Jika permintaan yang diminta oleh pelanggan dikenali atau infomasi yang diminta telah

sedia ada, kemudian pelayan akan mengirimkan data atau informasi tersebut ke pelanggan. Pada umumnya, pelayan hanya menunggu permintaan dari pelanggan. Pengaturcaraan pelanggan - pelayan menggunakan Java dapat dilaksanakan dengan mudah kerana Java telah menyediakan operasi - operasi yang diletak dalam kelas- kelas. Secara umum, kelas untuk pengaturcaraan rangkaian berada pada paket java.net.

Salah satu kaedah pengaturcaraan adalah menggunakan Socket. Socket menghubungkan antara proses yang satu ke proses yang lain dalam suatu rangkaian. Socket dalam Java mempunyai dua stream iaitu *input stream* dan *output stream*. Suatu proses mengirim data ke proses lain dengan menulis pada *output stream*. Suatu proses membaca data yang ditulis oleh proses lain dengan membaca input dari *input stream*. Dengan demikian ada satu socket dalam setiap komputer.



Rajah 5.5 : Model Rangkaian



Rajah 5.6 : Model Rangkaian Menggunakan Socket

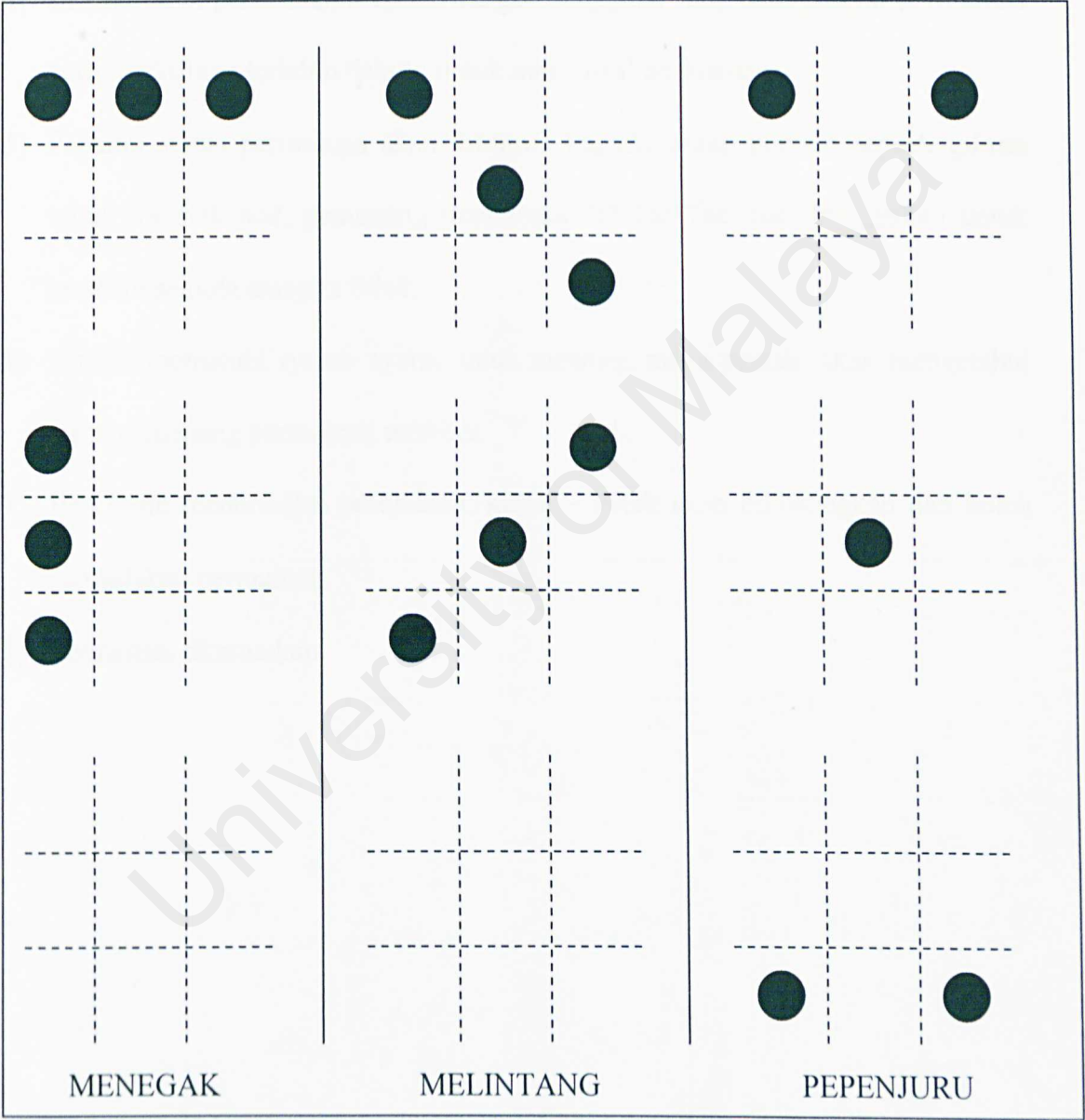
5.2.2 Rekabentuk Permainan

Permainan 3D Tic Tac Toe adalah merupakan permainan minda dan hanya menyediakan papan dan juga nod kepada pengguna. Pemain hanya perlu menggunakan tetikus sahaja untuk membuat tindakan. Di sini diterangkan cara permainan berjalan dan juga aliran gambarajah keseluruhan permainan yang akan dibangunkan.

Cara permainan 3D Tic Tac Toe

- 1) Pemain perlu memilih samada ingin bermain dengan komputer atau dengan pengguna lain.
- 2) Pemain akan diberi pilihan dua nod iaitu bulat ataupun pangkah.
- 3) Jika pemain memilih permainan berlawanan sesama pemain (rangkaian), permainan adalah secara bergilir dimana pemain A perlu meletakkan terlebih dahulu nod pada kotak yang diingini dalam 3 papan yang di sediakan.
- 4) Kemudian pemain B pula akan meletakkan nod nya pada kotak dalam 3 papan itu juga dan diikuti seterusnya secara bergilir sehingga habis kotak – kotak pada papan tersebut.
- 5) Pemain yang berjaya meletakkan 3 nod pada satu barisan samada dalam keadaan menegak, melintang atau pepenjuru di anggap sebagai pemenang.
- 6) Jika pemain memilih permainan berlawanan dengan komputer, permainan adalah secara bergilir.
- 7) Pemain yang berjaya mendapat lebih banyak barisan pada kotak – kotak tersebut samada dalam keadaan menegak, melintang atau pepenjuru di anggap sebagai pemenang.

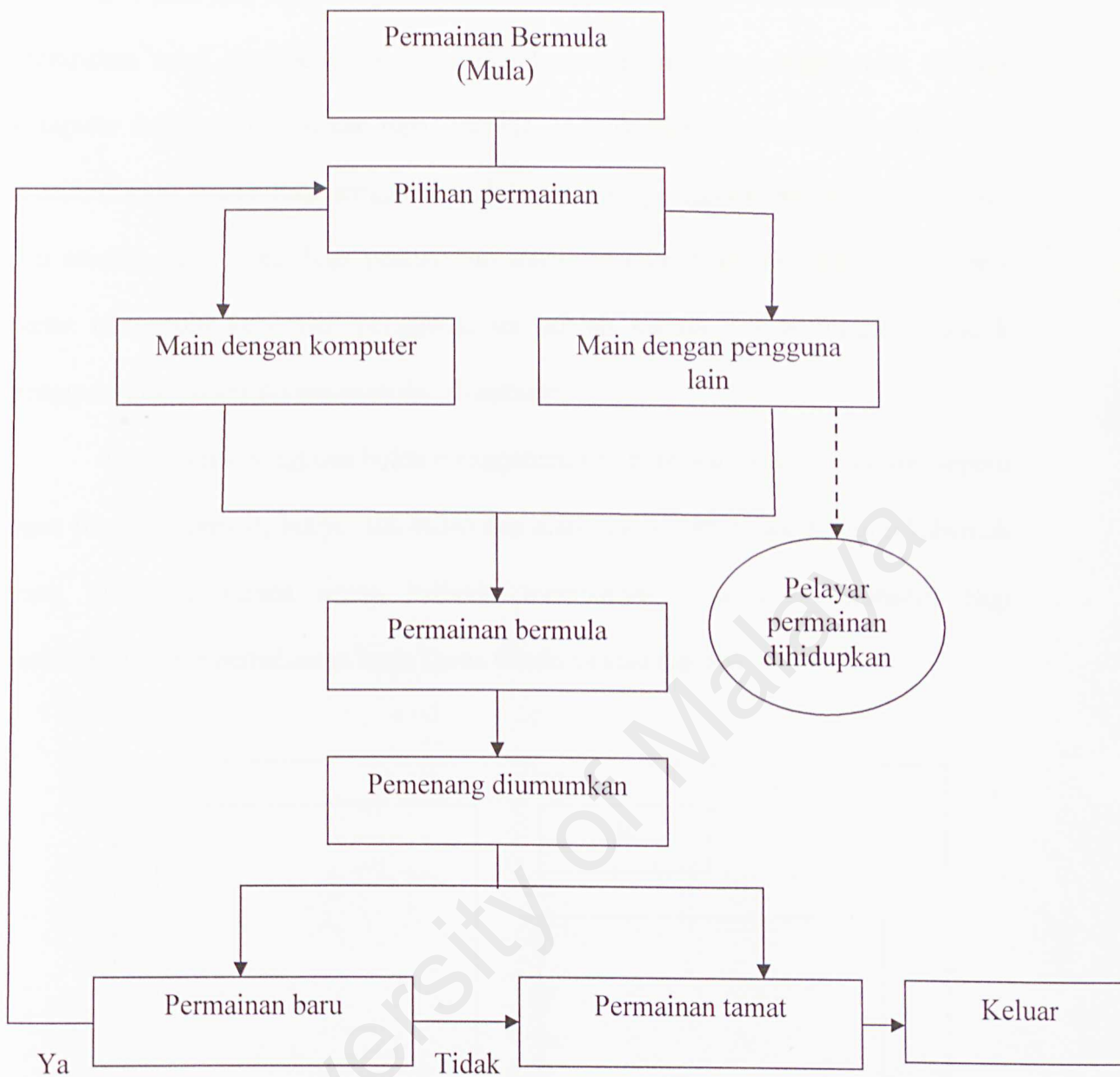
Contoh di bawah menunjukkan bagaimana syarat menang permainan 3D Tic Tac Toe. Nod bulat dijadikan contoh untuk menunjukkan susunan barisan yang diperlukan sebagai syarat memenangi permainan. Terdapat 3 keadaan iaitu menegak, melintang dan pepenjuru. Pada keadaan pepenjuru ianya boleh bermula dari kanan ke kiri ataupun kiri ke kanan.



Rajah 5.7: Cara memenangi permainan 3D Tic Tac Toe

Cara memulakan permainan 3D Tic Tac Toe

- 1) Pengguna perlu membuka Command Prompt untuk memulakan permainan.
- 2) Kemudian pemain membuat pilihan samada ingin bermain perseorangan (berlawan dengan komputer) ataupun bermain dengan pengguna yang lain.
- 3) Setelah pemilihan dibuat, permainan akan diaktifkan.
- 4) Jika memilih permainan bermain dengan pengguna yang lain, pelayar permainan perlu diaktifkan terlebih dahulu untuk mengawal permainan.
- 5) Paparan status permainan akan ditunjuk kepada setiap pemain seperti giliran untuk meletak nod, pemenang permainan 3D Tic Tac Toe dan pilihan untuk bermain semula ataupun tidak.
- 6) Setelah mematuhi syarat- syarat untuk menang, maka pemain akan mengetahui siapa pemenang permainan tersebut.
- 7) Jika ingin meneruskan permainan, kotak – kotak akan dikosongkan dan boleh memulakan permainan.
- 8) Permainan ditamatkan.



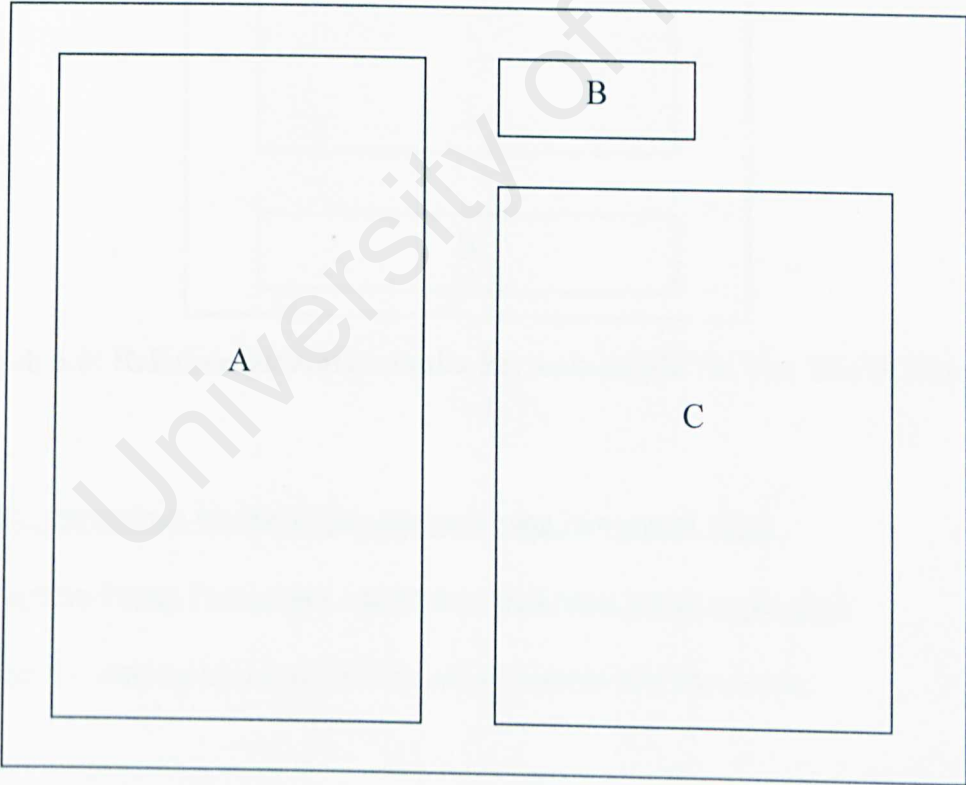
Rajah 5.8 : Gambarajah Aliran Permainan 3D Tic Tac Toe

5.3 Rekabentuk Antaramuka

Kebanyakan antaramuka pengguna mempunyai komponen visual contohnya laman web dan juga antaramuka pengguna bergrafik (GUI). Memahami asas rekabentuk

boleh membantu kita dalam merekabentuk antaramuka tersebut. Antaramuka pengguna merupakan salah satu bahagian yang membenarkan pengguna berinteraksi dengan komputer dalam melaksanakan tugas mereka. Ia juga adalah bahagian penting dalam sistem, dimana antaramuka pengguna ditujukan kepada pengguna dan bukannya kepada diri sendiri. Agak sukar bagi pembangun sistem mendapat antaramuka yang benar - benar memenuhi keperluan pengguna, ini adalah kerana semua tindak - tanduk pengguna amat sukar dikenalpasti dan ditentukan.

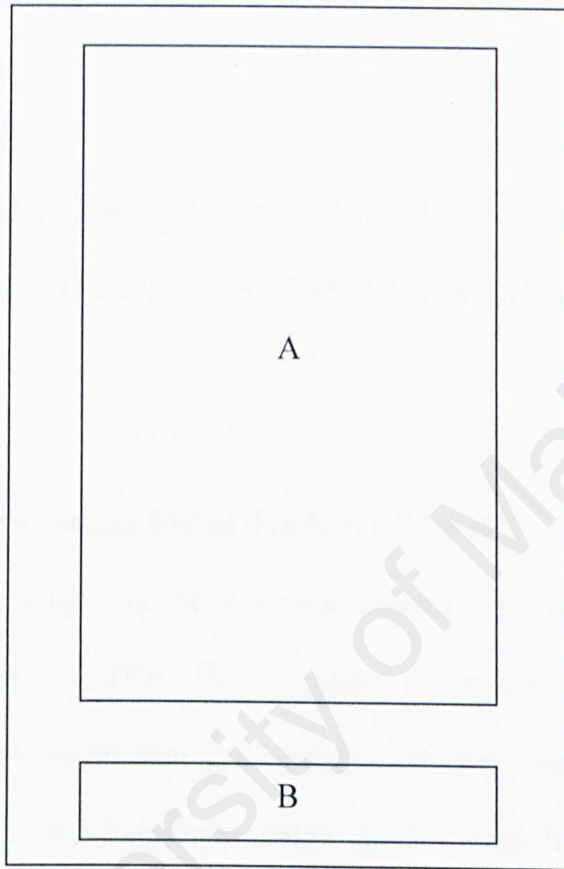
Antaramuka pengguna boleh menggabungkan pelbagai variasi teknologi seperti agen (bots), hypertext, bunyi, 3D, video dan alam maya. Ianya merupakan rekabentuk yang mencabar kerana setiap individu mempunyai gaya yang tersendiri bagi maklumbalas dan pemahaman kerja (guna windows atau arahan).



Rajah 5.9 : Rekabentuk Antaramuka Permainan 3D Tic Tac Toe (pertama)

Penerangan permainan berlawan dengan komputer :

- A. Paparan Papan Permainan – peletakan nod bulat ataupun pangkah
- B. Butang Main Semula – samada ingin bermain semula ataupun tidak
- C. Menu – status pemain, keputusan dan juga arahan permainan



Rajah 6.0: Rekabentuk Antaramuka Permainan 3D Tic Tac Toe (Kedua)

Penerangan permainan berlawan dengan pengguna lain (rangkaian) :

- A. Paparan Papan Permainan – peletakan nod bulat ataupun pangkah
- B. Menu – status permainan, giliran setiap pemain dan keputusan.

5.4 Rekabentuk Program

Di dalam rekabentuk program, ia menekankan kepada penggunaan pengaturcaraan piawai, prosedur dan panduan di dalam pemilihan struktur data serta algoritma. Selain itu, rekabentuk ini menyediakan dokumentasi sistem seperti dokumentasi program, operasi pengguna (memudahkan kerja penyelenggaraan sistem).

Huraian mengenai Socket:

Socket merupakan kombinasi antara alamat IP dan nombor port. Terdapat 4 jenis Socket (Socket class) pada Java iaitu Socket, ServerSocket, Datagram Socket dan Multicast Socket.

5.4.1 Pemrograman dengan Socket dan ServerSocket

Pemrograman rangkaian menggunakan Socket class dan ServerSocket class merupakan pemrograman TCP. Socket class digunakan pada pelanggan dan ServerSocket class pula digunakan pada pelayan. Socket class mempunyai 4 jenis arahan di mana setiap satu fungsinya adalah untuk mewakili alamat pelayan yang hendak dituju.

Socket class mempunyai operasi - operasi *getInputStream()*, *getOutputStream()* dan *close()* masing - masing digunakan untuk mengambil data, mengeluarkan data melalui Socket dan mengakhiri suatu hubungan. Operasi lain yang disediakan adalah operasi untuk mendapatkan informasi mengenai port yang digunakan pada sambungan 'hos local' dan 'hos remote'. ServerSocket digunakan untuk melayani permintaan dari pelanggan. Ia mempunyai 2 arahan iaitu 'port local' yang digunakan untuk menunggu

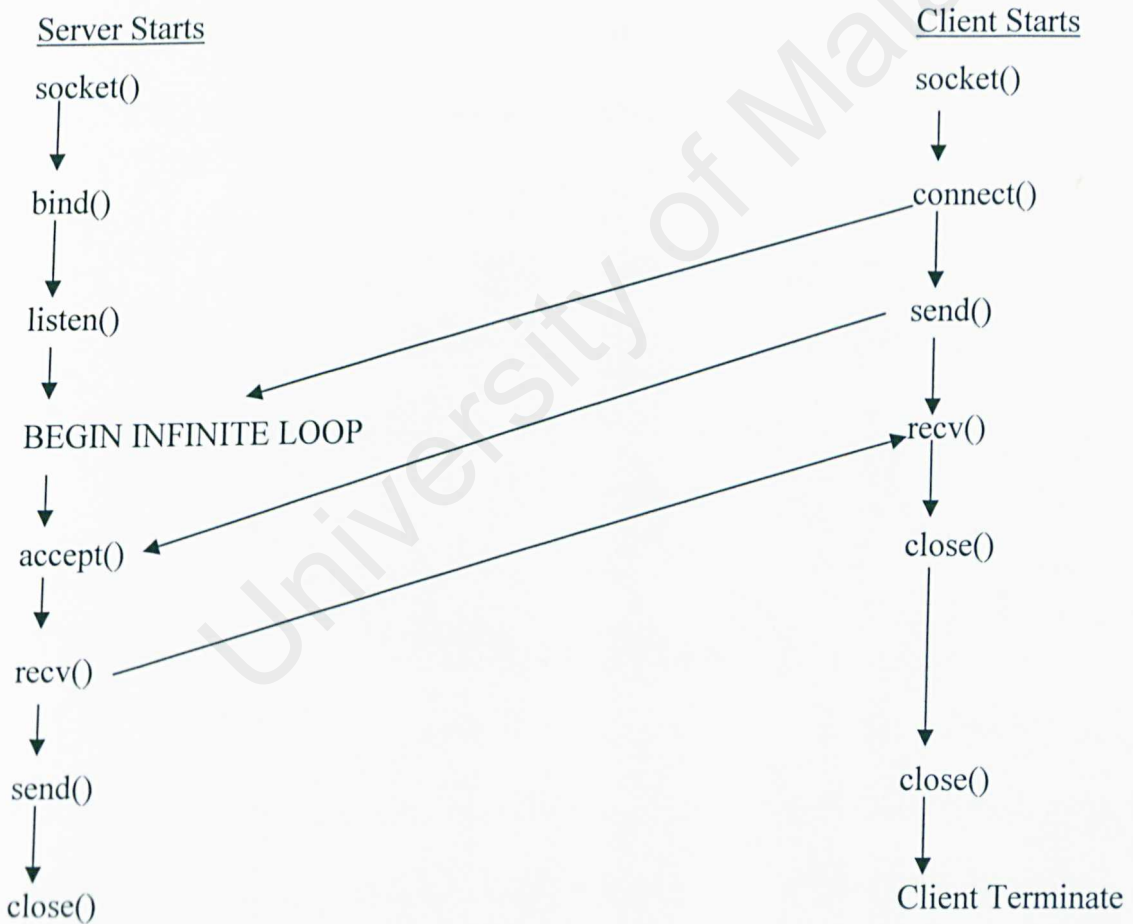
permintaan sambungan dan arahan untuk mengawal masa. `ServerSocket` mempunyai operasi penting berkaitan dengan pertukaran data iaitu `accept()` yang digunakan untuk memberitahu `Socket`(pelanggan) bahawa sambungan diterima dan `close()` digunakan untuk menghentikan operasi menunggu sambungan dari pelanggan. Selain itu juga, terdapat operasi untuk mendapatkan informasi nama host dan nombor port yang digunakan untuk operasi menunggu sambungan.

5.4.2 Datagram Socket

Pemrograman menggunakan datagram adalah pemrograman yang tidak berorientasi pada sambungan, sebagaimana penggunaan `UDP` (`User Datagram Protocol`). Terdapat dua class yang disediakan oleh Java iaitu `DatagramPacket` dan `DatagramSocket`. `DatagramPacket` adalah paket data dalam byte array di mana ia akan dipindahkan dari satu komputer ke komputer lain, manakala `DatagramSocket` adalah `Socket` yang mengirim dan menerima `DatagramPacket` melalui rangkaian. `DatagramPacket` class mempunyai dua arahan yang digunakan untuk ukuran yang diterima dalam bentuk array serta menyimpannya. Bagi arahan yang lain digunakan untuk menganalisis `DatagramPacket` yang akan dikirim. Kedua - dua arahan memerlukan informasi yang sama termasuk alamat pelanggan dan nombor port. `DatagramPacket` class ini mempunyai 4 operasi yang digunakan untuk mendapatkan data, panjang data, alamat dan nombor port. `DatagramSocket` mempunyai 2 arahan di mana pengaturcara dapat menentukan sendiri nombor port yang digunakan. Dua operasi yang penting adalah `send()` dan `receive()` yang masing - masing digunakan untuk mengirim dan menerima paket data ke/dari host dan nombor port tertentu.

5.4.3 Komponen utama dalam pemrograman pelanggan – pelayan

- Socket
- Bind
- Listen
- Connect
- Accept
- Recieve
- Send



Rajah 6.1 : Aliran Pemrograman Pelanggan - Pelayan

Bab 6: Pengkodan

University of Malaysia

Bab 6 : Pengkodan

6.1 Pengenalan Pengkodan

Dalam fasa ini, masa yang diambil adalah paling lama berbanding dengan fasa - fasa lain kerana pada fasa inilah letaknya proses - proses rekaan dan pengubahsuaian yang melibatkan fungsi - fungsi tertentu yang berkaitan dengan sistem yang dibangunkan. Kod program yang dibina akan dikompil oleh pengkompil bagi menyemak samada terdapat kesalahan pada kod program tersebut ataupun tiada, jika tiada sebarang kesalahan kod program akan dilarikan dan dilaksanakan. Tindakan ini akan dilakukan oleh JDK (Java Development Kit). Terdapat faktor - faktor kekangan terhadap pembangunan sistem yang perlu dipertimbangkan, antaranya ialah:

- i. Faktor masa dan tenaga yang terhad bagi fasa ini.
- ii. Faktor kemahiran dan juga pengetahuan mengenai kod program java pada pembangun sistem itu sendiri.
- iii. Faktor bahan rujukan untuk menghasilkan permainan dalam bentuk kod program java.

6.2 Pendekatan Pengkodan

Pembangun sistem menghadapi masalah ketika membentuk program kepada bentuk tindakan yang dikehendaki. Ini adalah kerana secara konsepnya ia mudah dibentuk tetapi secara praktikal ia tidaklah semudah yang dijangkakan. Keupayaan pembangun sistem diuji, di mana setiap peristiwa (*event*) yang dijangkakan ke atas sistem memerlukan penulisan secara manual (melalui pengetahuan pembangun). Pembangun juga perlu menggunakan kreativiti dan pengetahuan sedia ada dalam

mewujudkan peristiwa - peristiwa agar bentuk tindakan yang terhasil adalah relevan dengan setiap peristiwa yang diwujudkan. Penggunaan kod program java juga perlu dipelajari terlebih dahulu seperti kelas – kelas yang akan digunakan, jenis – jenis operasi aritmetik dalam pembinaan algoritma dan juga teknik penghasilan imej yang sesuai untuk permainan yang akan dibangunkan. Rujukan ke atas sistem sedia ada membantu pembangun sistem mengurangkan bebanan dalam memperoleh fungsi - fungsi yang sepatutnya digunakan untuk sesuatu arahan dan juga algoritma bagi permainan 3D Tic Tac Toe.

6.2.1 Kaedah - Kaedah Pengkodan

Kaedah – kaedah yang digunakan dalam pengkodan :

➤ Carta Aliran

Carta aliran merupakan wakil keseluruhan atau sebahagian dari algoritma dalam bentuk grafik. Ia dilukis dengan menggunakan simbol – simbol khas yang disambungkan dengan anak panah dipanggil garis aliran. Carta aliran sama pentingnya dengan penggunaan pseudocode terutama dalam menunjukkan bagaimana struktur kawalan berfungsi.

➤ Algoritma

Algoritma adalah prosedur untuk menyelesaikan masalah dalam bentuk arahan yang akan dijanakan dalam kod program. Ia merupakan suatu kaedah matematik yang diolah ke dalam susunan arahan – arahan dimana arahan tersebut dibentuk dengan menggunakan simbol – simbol khas dari operator aritmetik.

➤ Pseudocode

Pseudocode adalah bahasa buatan dan tidak formal yang membantu pembangun sistem membina algoritma. Ia adalah lebih kurang sama dengan bahasa harian dan bukan bahasa komputer yang sebenar. Pseudocode ditulis sepenuhnya dengan menggunakan huruf dan hanya mengandungi pernyataan yang boleh dijanakan. Pengisytiharan tidak termasuk dalam pseudocode.

➤ Jadual Keputusan

Jadual keputusan menyenaraikan keputusan yang diperolehi daripada algoritma yang telah dibentuk. Ia memudahkan pembangun sistem dalam melihat keseluruhan fungsi kod program yang dibina.

6.2.2 Format Pengkodan

Penerangan awal akan dilakukan pada kod program, ianya berdasarkan kepada kelakuan sistem apabila kod ini dilarikan. Penerangan ini juga akan dapat membantu meningkatkan kebolehbacaan sumber – sumber kod agar mudah dilakukan pengubahsuaian oleh pembangun sistem sekiranya perlu.

Pengkodan dimulakan dengan mengimport kelas java yang telah sedia ada dalam perpustakaan java. Ianya boleh dilakukan dengan menggunakan pernyataan *import*. Kelas – kelas java tersebut mempunyai sokongannya yang tersendiri contohnya *java.applet* untuk pelayar web, *java.net* untuk rangkaian, *javax.swing* untuk antaramuka grafik, *java.io* untuk capaian fail.

Selepas fungsi import, penentuan kelas untuk kod program dihasilkan dimana kelas adalah suatu objek yang menakrifkan atribut – atribut dan operasi – operasi bagi objek tersebut. Kelas – kelas yang telah ditentukan akan mendefinisikan atribut (keadaan bagi sesuatu objek) dan juga operasi atau kelakuan sesuatu program. Pengisytiharan pembolehubah dalam kod program adalah mengikut kepentingan setiap pembolehubah itu sendiri. Terdapat 3 jenis kawalan capaian yang boleh digunakan untuk pembolehubah tersebut :

➤ *Kawalan capaian private*

Ia menghadkan keatas pembolehubah ahli atau kepada pengisytiharan kelas yang menakrifkannya.

➤ *Kawalan capaian public*

Ia tidak menghadkan capaian keatas pembolehubah ahli atau metod. Ini bermakna capaian boleh dilakukan di mana – mana tempat di dalam aturcara.

➤ *Kawalan capaian protected*

Ia berkait rapat dengan pewarisan. Unsur yang protected boleh dicapai oleh kelas yang menakrifkannya dan juga subkelas.

Di dalam pengkodan juga boleh ditulis komen – komen dimana ianya adalah untuk tujuan dokumentasi. Biasanya digunakan untuk menerangkan tujuan program dan bahagian – bahagian tertentu di dalamnya. Ianya dihiraukan oleh pengkompil semasa kompilasi dan tiada kesan kepada perlaksanaan program. Komen boleh ditulis dengan menggunakan aksara `/** dokumentasi */` , `/* dokumentasi */` dan `//` .

6.3 Pengkodan Permainan 3D Tic Tac Toe

Kaedah pengkodan permainan dibahagikan kepada dua bahagian iaitu bermain dengan komputer dan juga bermain sesama pengguna (rangkaian). Bagi permainan dengan komputer, algoritma terbahagi kepada 3 iaitu menang, halang dan rawak. Penggunaan operasi aritmetik (for .. if) untuk menukarkan kaedah matematik ke dalam bentuk arahan. Bagi permainan sesama pengguna (rangkaian), permainan dikawal oleh pelayan permainan (server) dimana ianya berfungsi dalam menentukan giliran setiap pemain dengan menggunakan kaedah *socket*, algoritma permainan menggunakan operasi aritmetik (|| , != , && , ==). Pelanggan permainan pula menyediakan pilihan samada pemain ingin meneruskan permainan ataupun tidak dengan menggunakan JOptionPane dan penentuan status pemain (if else).

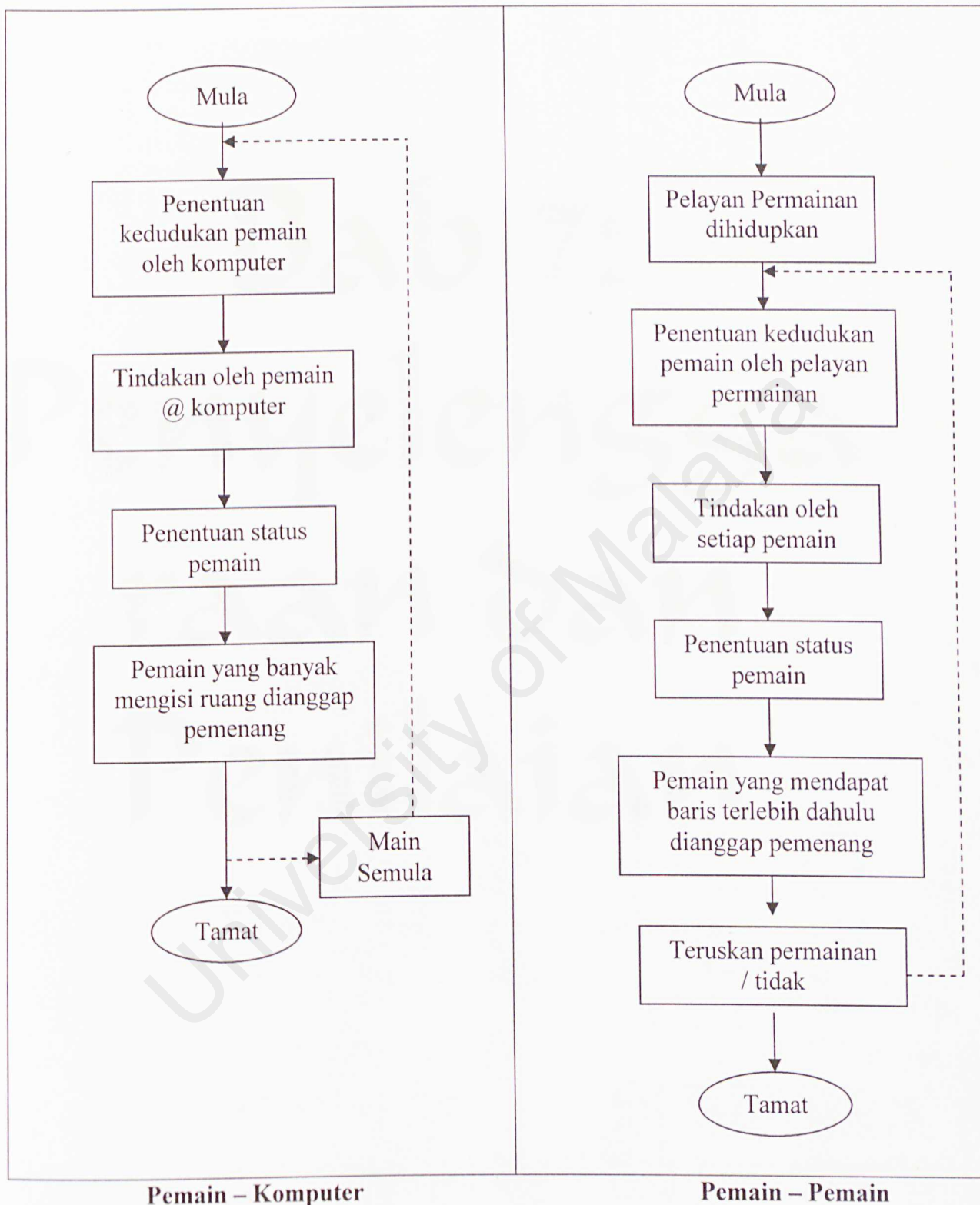
Cara untuk memenangi permainan 3D Tic Tac Toe ini dibahagikan kepada dua konsep iaitu :

➤ Konsep Banyak – Menang

Setelah selesai tindakan yang dilakukan oleh pemain atau komputer, maka keputusan akan diketahui dengan melihat jumlah baris yang telah diperolehi oleh nod X dan juga nod O. Jumlah yang paling banyak akan dianggap sebagai pemenang.

➤ Konsep Kalah – Mati

Pemain akan bermain mengikut giliran masing – masing, dan berusaha untuk menghalang antara satu sama lain dari mengisi satu barisan, bagi pemain yang mendapat satu barisan terlebih dahulu dianggap sebagai pemenang.



Rajah 6.1 : Carta Aliran Permainan 3D Tic Tac Toe

Bab 7: Penyelengs- raan dan Penilaian

Bab 7 : Penyelenggaraan dan Penilaian

7.1 Penyelenggaraan Sistem

Pada bab ini, perbincangan difokuskan kepada keperluan penyelenggaraan sistem dan bagaimana penyelenggaraan sistem ini dapat dilakukan apabila keperluan fungsian semakin berubah. Ini adalah bertujuan untuk memberikan panduan dan pemahaman kepada pengguna serta pembangun sistem itu sendiri untuk menjalankan penyelenggaraan terhadap sistem yang dibangunkan agar ianya tidak menjejaskan pengoperasian sistem secara keseluruhan.

7.1.1 Keperluan Penyelenggaraan

Secara amnya sesebuah sistem harus diselenggarakan dari masa ke semasa untuk memastikan sistem beroperasi pada tahap yang optimum. Ianya juga perlu dalam mengawasi setiap perubahan yang boleh menjejaskan operasi sistem tersebut. Oleh itu, sistem akan sentiasa berada dalam keadaan yang stabil dan terkawal.

7.1.2 Metodologi Penyelenggaraan Sistem

Sistem ini dapat diselenggarakan dengan kaedah yang terdapat di bawah :

- **Perubahan Teknologi**

Java membenarkan versi lama beroperasi di dalam versi baru dan juga sebaliknya. Tetapi terdapat juga kod – kod dan objek - objek tertentu yang tidak sebaliknya. Ini menunjukkan perubahan versi perisian tidak memberi kesan kepada sistem. Perubahan sistem pengoperasian juga tidak menjejaskan pengoperasian sistem ini kerana ia dipengaruhi oleh perubahan teknologi.

7.1.3 Pelan Pengembalian Bencana

Pelan pengembalian bencana bertujuan untuk menyediakan sandaran kepada pengoperasian sistem sekiranya berlaku bencana. Kandungan asas sistem perlu disalin ke dalam storan lain seperti media storan 'back-up' (back – up CD-RW, disket, cakera keras yang lain, back – up tape dan sebagainya). Ini adalah bertujuan mengelakkan sebarang kehilangan data – data yang telah dibina dan dihasilkan terlebih dahulu. Setidaknya jika berlaku sebarang bencana, sistem masih boleh diperbaiki dan digunakan melalui data – data terdahulu dan pembangun sistem tidak perlu membina semula keseluruhan sistem tersebut. Selain itu juga, dokumentasi yang telah dibuat tidak perlu dihasilkan semula bagi mengelakkan pembaziran masa kerana sememangnya masa adalah terhad.

7.2 Penilaian Sistem

Salah satu cara yang baik untuk membangunkan sesebuah sistem adalah dengan melibatkan fasa penilaian sistem di dalam kitar pembangunan sistem tersebut. Fasa penilaian bertujuan agar pembangun sistem dapat menganalisa sejauh manakah kejayaan dan keberkesanan sistem yang dibangunkan samada ia mencapai objektif ataupun tidak. Biasanya pembangun sistem akan terima maklumbalas daripada pengguna tentang sistem mereka kerana pengguna akan menilai sesebuah sistem samada ianya memenuhi kehendak pengguna ataupun tidak.

7.2.1 Pencapaian Objektif

Secara amnya, sistem ini berjaya mencapai objektif yang digariskan semasa fasa analisis dijalankan. Namun terdapat fungsi yang rumit dan kompleks untuk direalisasikan oleh pembangun sistem. Oleh sebab itu, pembangun sistem telah membangunkan fungsian alternatif lain sebagai gantian. Objektif utama sistem ini adalah supaya pengguna dapat bermain sebuah permainan menggunakan rangkaian (multi – pemain), dan ianya berhasil dan telah teruji.

7.2.2 Kekuatan Sistem

- i. Antaramuka yang menarik dan mesra pengguna
 - Sistem ini mampu digunakan oleh sesiapa sahaja dan mudah untuk difahami kerana status pemain akan dipaparkan contohnya samada pengguna ingin bermain semula ataupun tidak, paparan untuk menunjukkan giliran setiap pemain dan juga samada menang ataupun kalah. Penggunaan warna juga adalah sesuai dengan kehendak pengguna kerana ianya adalah daripada permintaan pengguna yang telah mencuba terlebih dahulu sistem ini.
- ii. Mudah Digunakan
 - Sistem ini mudah digunakan kerana pengguna akan mengambil masa yang singkat untuk membiasakan diri dengan sistem ini kerana ia mempunyai kekunci yang terhad. Kebanyakan tindakan pengguna

hanyalah bergantung kepada penggunaan tetikus dan juga idea pemikiran untuk memenangi permainan 3D Tic Tac Toe ini.

7.2.3 Kekangan Sistem

➤ Kelenturan Sistem

Kelenturan Sistem kurang, dimana peraturan – peraturan telah ditetapkan dalam algoritma permainan dan tidak boleh ditukar kerana ia merupakan implementasi ketika fasa pengkodan. Perubahan hanya boleh dilakukan dengan mengubahsuai kod program yang telah dibentuk. Ianya adalah suatu perkara yang rumit dan hanya boleh dilakukan oleh pembangun sistem sahaja.

7.3 Penambahan pada masa hadapan

Sistem yang dibangunkan masih banyak yang boleh ditambah supaya ianya dapat berfungsi dengan lebih baik dan menarik. Antara penambahan yang boleh dilakukan adalah seperti berikut :

- i. Menyimpan data permainan dan boleh dicapai mengikut kekuncian carian.
- ii. Penambahan terhadap video dan juga animasi.
- iii. Membenarkan pengguna untuk melakukan pengeditan atau pertukaran terhadap sesuatu fungsian sistem yang boleh ubah seperti perubahan dalam antaramuka grafik permainan dan juga pengaturcaraan permainan.

Bab 8: Pengujian

University of Malaya

Bab 8 : Pengujian

8.1 Peringkat Pengujian Sistem

Pengujian diperlukan untuk memastikan supaya sistem dapat beroperasi seperti yang dikehendaki. Jika terdapat kesilapan - kesilapan maka pembetulan akan dibuat dan biasanya melibatkan beberapa perubahan kecil. Sistem diuji supaya bebas ralat terutamanya ralat masa larian yang tidak dapat dikesan semasa pengkompilan program.

Pengujian ke atas sistem boleh dibahagikan kepada 3 peringkat :

- 1) Pengujian Unit
- 2) Pengujian Keseluruhan Sistem
- 3) Pengujian Pemeriksaan

Dalam peringkat pertama, setiap modul akan disemak secara berasingan supaya ia bebas ralat. Pada peringkat kedua kesemua modul disambungkan antara satu sama lain dan penyemakan ke atas keseluruhan sistem dilakukan. Pada peringkat ketiga pula, ujian dilakukan terhadap pengguna lain untuk mengesan ralat yang tidak dapat dikesan oleh pembangun sistem.

Peringkat ketiga ini dapat memastikan pengujian yang dilakukan adalah adil dan seimbang kerana ianya melibatkan individu - individu di luar pembangunan sistem. Peringkat ini juga sangat penting kerana melalui pandangan - pandangan pengguna, pembangun dapat memperbetulkan program yang mungkin mempunyai beberapa kelemahan yang sepatutnya dielakkan seperti arahan pada menu yang sukar difahami, ikon yang sukar difahami atau terdapat ralat yang mungkin berlaku apabila pengguna

menekan kunci yang tidak sepatutnya ditekan dalam keadaan tertentu. Kesemua masalah yang terdapat dalam peringkat yang ketiga biasanya tidak disedari atau sukar dijangka melainkan jika melibatkan pengguna lain untuk menguji sistem yang dibangunkan.

8.1.1 Pengujian Unit

Setiap prosedur dan fungsi di dalam antaramuka sistem diuji secara berasingan dengan data - data bagi memastikan ianya berfungsi dengan betul. Data - data ujian dimanipulasikan dengan menguji semua syarat - syarat dalam satu segmen kod untuk memastikan ianya bena - benar boleh berfungsi dengan baik seperti yang dikehendaki. Semua segmen kod yang diuji dengan baik menjamin kepantasan dan kebolehpercayaan sistem serta memudahkan ujian - ujian seterusnya (ujian integrasi sistem) ke atas sistem dijalankan.

8.1.2 Pengujian Keseluruhan Sistem

Sekumpulan modul antaramuka sistem yang diintegrasikan diuji untuk melihat kelakuan dan tindak balas ke atas modul lain dan komunikasi pengguna dengannya. Ujian juga menitikberatkan kesesuaian rekabentuk antaramuka dengan kelakuan sistem yang sepatutnya. Sebarang ralat dalam integrasi sistem yang disebabkan ketidakserasian modul - modul di antara satu sama lain diperbetulkan dengan melakukan ujian stuktur sistem.

Antara ujian- ujian yang telah dilakukan dalam fasa ini ialah:

➤ *Ujian Integrasi Data*

- Mengesahkan data - data bertindak dengan betul semasa digunakan di dalam sistem.

➤ *Ujian Regrasi*

- Menentukan sebarang ralat atau kesan sampingan yang terhasil ketika membetulkan ralat.

➤ *Ujian Keselamatan.*

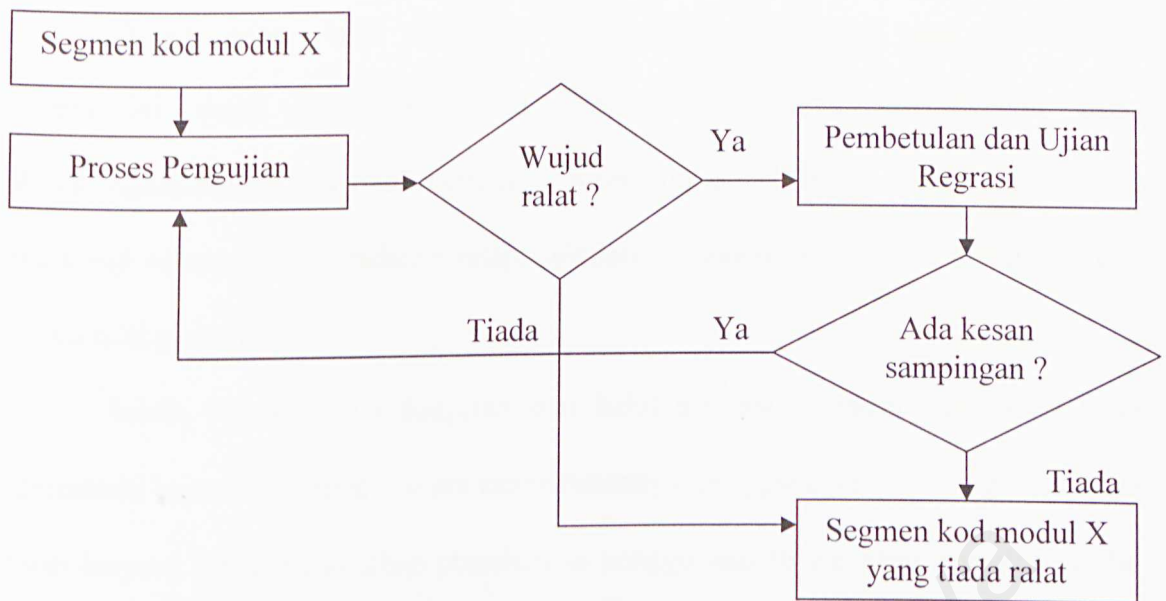
- Ujian capaian terhadap keselenggaraan.

8.1.3 Pengujian Penerimaan

Di dalam ujian ini, pengguna dikehendaki menggunakan sistem ini tanpa sebarang bantuan dari pembangun sistem. Ini adalah bertujuan untuk menguji kefahaman pengguna ke atas fungsian sistem dan tahap mesra pengguna yang diterapkan ke dalam sistem.

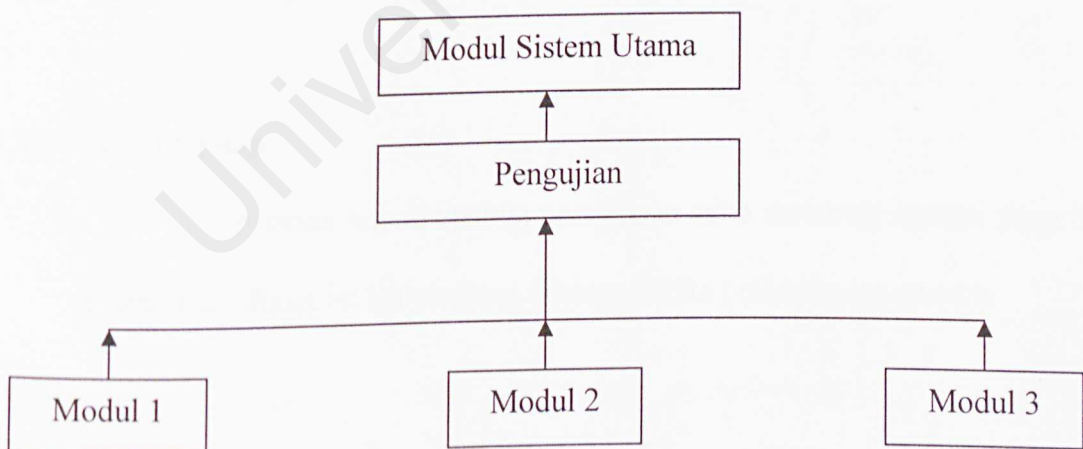
8.2 Teknik Pengujian Sistem

Teknik yang digunakan adalah mengikut peringkat ujian yang telah ditetapkan. Pada peringkat pengujian unit, teknik kotak putih (*white box*) telah digunakan untuk menentukan sebarang ralat mengikut rajah skema di bawah:



Rajah 8.1 : Skema Ujian Unit

Di peringkat sistem, teknik integrasi bawah – atas (*bottom – up*) digunakan dengan ujian dimulakan pada modul - modul aras paling bawah dan membinanya dari situ seperti skema pada rajah di bawah:



Rajah 8.2 : Skema Ujian Sistem Teknik Bawah – Atas

Dengan ujian - ujian yang telah dilaksanakan, ralat - ralat yang wujud dalam sistem ini dapat dikenalpasti dan kemudiannya diselesaikan dengan baik. Walaubagaimanapun, ini tidak bermakna sistem ini adalah betul - betul bersih atau tiada lagi sebarang ralat padanya tetapi sekurang- kurangnya ini dapat mengurangkan risiko ralat pada sistem.

Selain itu, kesemua fungsian dan kelakuan sistem dapat dipastikan telah memenuhi keperluan sistem. Secara keseluruhannya pengguna yang menguji sistem ini telah berpuas hati dengan tahap pemahaman penggunaan sistem yang mudah (melalui Ujian Penerimaan).

8.3 Ralat Ketika Pengujian

8.3.1 Ralat Kompilasi

Ralat ini terjadi dalam pembinaan kod yang sah disebabkan kesilapan pada kata kunci ataupun tertinggalnya tanda - tanda penting.

8.3.2 Ralat Larian

Ralat larian terjadi apabila pernyataan cuba membuat operasi yang tidak munasabah. Ralat ini hanya dapat dikesan ketika proses larian atucara.

8.3.3 Ralat Logik

Ralat ini berlaku apabila sistem tidak menghasilkan output seperti yang dikehendaki.

Bab 9 : Masalah Dan Cadangan

Bab 9 : Masalah dan Cadangan

9.1 Masalah yang dihadapi

Dalam proses menyiapkan permainan 3D Tic Tac Toe, terdapat beberapa masalah berbangkit yang terpaksa dihadapi oleh pembangun sistem dimana masalah tersebut perlu diatasi sebelum sistem dapat disiapkan. Masalah perlu dibincangkan supaya ianya tidak akan memberi kesan terhadap sistem yang dibangunkan. Di antara beberapa masalah yang dihadapi adalah seperti berikut :

9.1.1 Rujukan terhad, agak mahal dan sukar didapati

Memandangkan sistem saya ini adalah berkonsep sebuah permainan dan ianya adalah terlalu teknikal maka saya harus merujuk kepada sistem – sistem yang sedia ada untuk mendapatkan idea dan juga fungsi – fungsi yang digunakan. Kebanyakan sistem yang sedia ada menggunakan bahasa pengaturcaraan C++ dan Visual Basic. Ini menyebabkan saya mengambil masa yang agak lama berbanding dengan jadual perancangan yang telah dibuat dan ianya sedikit sebanyak menjejaskan tugas – tugas serta fasa – fasa lain seperti dokumentasi dan ciri – ciri tambahan yang sepatutnya dimasukkan ke dalam sistem.

9.1.2 Penggunaan arahan dalam bahasa pengaturcaraan Java

Arahan – arahan yang terdapat dalam bahasa pengaturcaraan Java sememangnya masih lagi asing pada diri saya walaupun saya telah mempelajari bahasa Java semasa proses pembelajaran saya di Universiti Malaya namun masa pembelajaran bahasa Java adalah singkat kerana ia merupakan kursus pendek dan saya mempelajarinya pada masa

yang sama saya mengambil latihan ilmiah tahap akhir II ini. Dalam membangunkan permainan ini, saya perlu banyak menggunakan fungsi – fungsi serta pembinaan algoritma yang sesuai dan dikehendaki. Tegasnya disini asas yang kukuh dalam bahasa pengaturcaraan khususnya bahasa Java amat perlu dalam membantu memahami dan mengolah arahan – arahan ke dalam bentuk kod program yang lebih menarik dan interaktif.

9.1.3 Pertukaran dalam fungsian sistem

Terdapat beberapa fungsian sistem yang tidak dapat disertakan disebabkan kekangan masa dan keupayaan dalam pengkodan yang rumit serta banyak kesilapan yang berlaku setelah ia cuba dilaksanakan. Semasa kertas cadangan sistem ini dibentangkan, penghasilan kotak perbualan seharusnya disertakan namun ianya tidak dapat dilaksanakan kerana kesilapan pada arahan – arahan rangkaian yang tidak dapat menghubungkan antara pemain untuk menggunakan kotak perbualan tersebut.

9.2 Cadangan

Bagi mengelakkan masalah seperti di atas berulang kepada pembangun sistem yang lain, beberapa cadangan telah dikemukakan untuk mengatasi masalah – masalah tersebut. Diantaranya adalah :

9.2.1 Rujukan terhad, agak mahal dan sukar didapati

Pembangun sistem seharusnya mengenalpasti terlebih dahulu bahan – bahan rujukan yang perlu dan mengumpulkan segala sumber – sumber rujukan yang berkaitan

sejak awal lagi. Pengumpulan sejak awal ini penting agar setiap fasa dapat dilaksanakan dan dokumentasi dapat dihasilkan dengan lengkap serta mengikut masa yang telah ditetapkan. Ini juga dapat mengelakkan gangguan lengahan pada fasa – fasa seterusnya. Pinjaman bahan rujukan yang berkenaan dengan sistem yang dibangunkan melalui perpustakaan Universiti ataupun awam dan juga orang perseorangan seperti meminjam buku – buku yang agak mahal serta sukar diperolehi.

9.2.2 Penggunaan arahan dalam bahasa pengaturcaraan Java

Pembangun sistem perlu bertanya kepada individu yang lebih mengetahui dan mahir dengan bahasa Java. Selain itu juga, pembangun sistem perlu merujuk di Internet tentang penggunaan bahasa Java dalam mencipta suatu permainan yang berkaitan dengan sistem yang akan dibangunkan. Ianya boleh dilaksanakan menerusi enjin pencarian seperti google, altavista, lycos dan lain – lain. Pembangun sistem juga boleh mencuba sebarang arahan – arahan sedia ada dan kemudian melihat output yang terhasil daripada arahan tersebut. Proses ini juga perlu dilakukan pada peringkat awal untuk proses pengkodan berjalan dengan lancar.

9.2.3 Pertukaran dalam fungsian sistem

Pembangun sistem seharusnya mencari contoh kod program sebagai rujukan. Kod program kotak perbualan tersebut adalah berkaitan dengan sistem yang akan dibangunkan. Kemudian, cuba mengaitkan kod program tersebut ke dalam kod program permainan yang dibangunkan dan melihat sendiri output yang terhasil.

9.3 Pengetahuan yang telah diperolehi

- i. Kemahiran dalam mencari dan mendapatkan maklumat serta mengolah maklumat – maklumat tersebut kepada bentuk yang dikehendaki oleh pembangun sistem.
- ii. Memperoleh beberapa teknik merancang dan membangunkan sesuatu sistem yang komersil.
- iii. Kemahiran dalam pengurusan masa dan juga projek.
- iv. Kemahiran berkomunikasi dengan orang perseorangan samada dikenali ataupun tidak dikenali.
- v. Menambahkan lagi pengetahuan sedia ada tentang bahasa Java dan memberi kefahaman serta kemahiran dalam penggunaan arahan – arahan bahasa Java.
- vi. Menimba pengetahuan mengenai teknik – teknik yang boleh digunakan dalam menyelesaikan sesuatu masalah.

9.4 Kesimpulan

Ternyata projek Latihan Ilmiah Tahap Akhir dapat mendekatkan para pelajar berkaitan teknik – teknik dalam membangunkan sesebuah sistem termasuk segala masalah yang dihadapi serta cara untuk mengatasi masalah tersebut. Dalam menyiapkan projek ini, saya telah mempelajari banyak pengetahuan baru berkaitan arahan – arahan dalam bahasa Java dimana ianya belum pernah saya gunakan sebelum ini.

Saya juga telah dapat merasai sendiri pengalaman membangunkan sebuah sistem secara individu dan juga mendapat bantuan daripada individu – individu lain selama setahun yang semestinya banyak memberi manfaat pada masa akan datang. Pengetahuan mengenai pengaturcaraan sememangnya perlu bagi pelajar yang mengambil jurusan Sains Komputer dan Teknologi Maklumat kerana ianya akan digunakan di tempat kerja kelak. Ternyata terdapat beberapa perbezaan pada Latihan Ilmiah Tahap Akhir I dan juga Latihan Ilmiah Tahap Akhir II, namun sistem yang telah dihasilkan adalah hampir sama mutunya berbanding dengan perancangan asal dan ianya boleh dikatakan menepati sasaran awal.

Semoga sistem yang dibangunkan ini menjadi titik tolak kepada individu lain untuk mencipta sebuah permainan yang lebih menarik dan juga lebih baik mutunya berbanding permainan sedia ada di pasaran sekarang, memandangkan bilangan pengaturcara tempatan yang menceburi bidang ini adalah sedikit.

Jika diberi peluang sekali lagi, saya amat berharap untuk menambahkan lagi pengetahuan saya tentang bahasa pengaturcaraan Java dan juga bahasa pengaturcaraan yang lain seperti Active Server Page (ASP), Hypertext Preprocessor (PHP), Java 3D, Visual Basic, JavaScript, perisian – perisian 3D dan lain – lain.

Lampiran

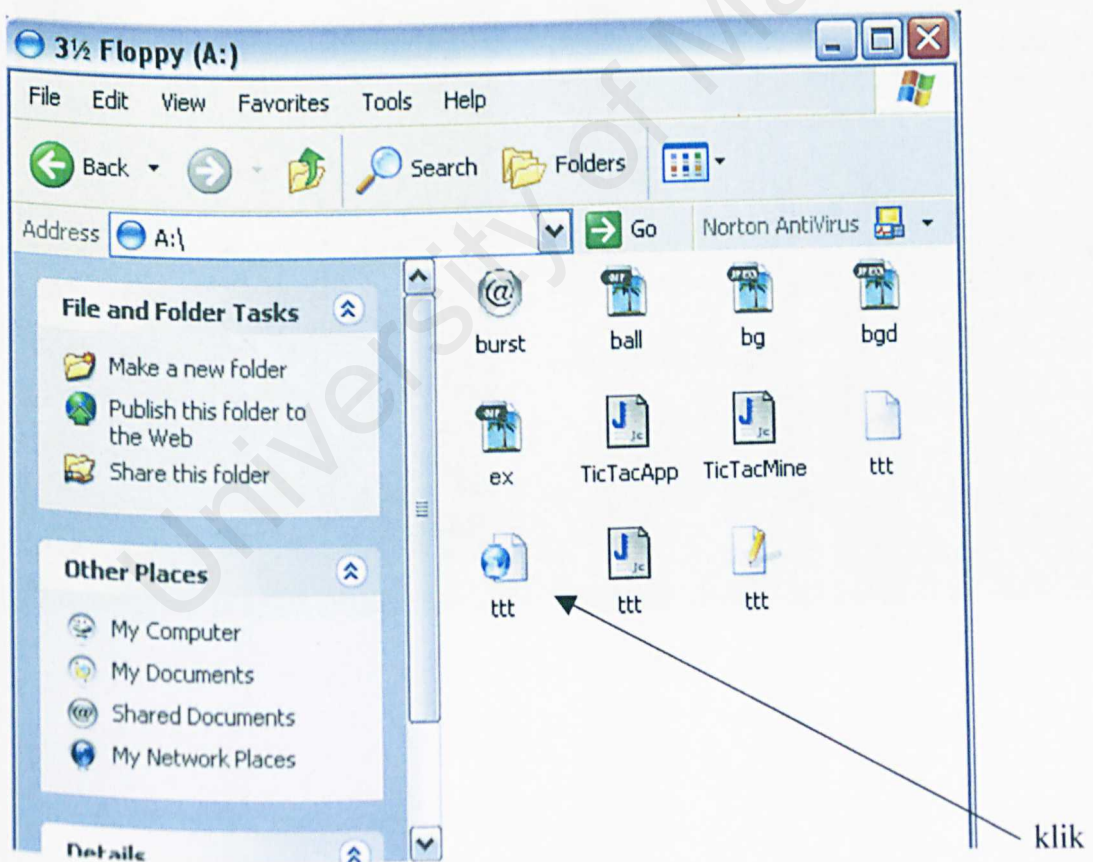
University of Malaya

Sistem Manual Pengguna

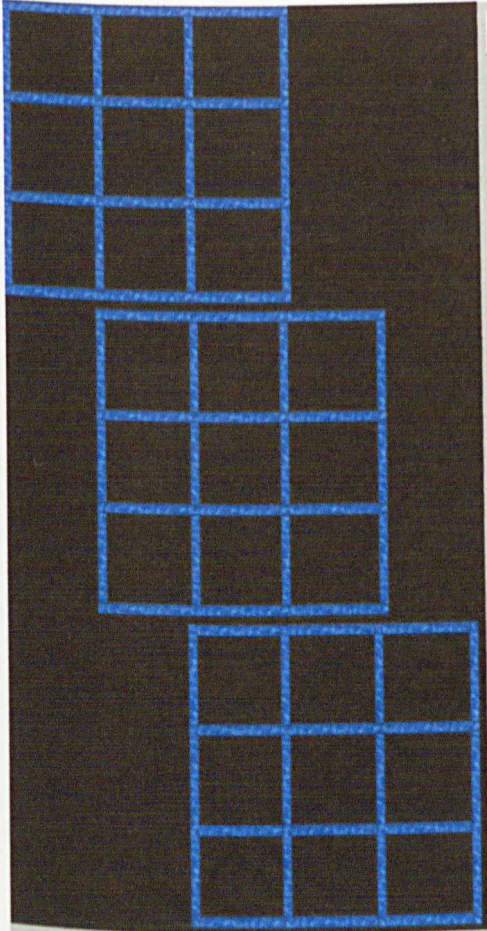
Di dalam bahagian lampiran, disertakan beberapa maklumat serta paparan yang berkenaan dengan sistem yang telah dibangunkan iaitu permainan 3D Tic Tac Toe. Di sini disertakan cara – cara untuk memulakan permainan 3D Tic Tac Toe.

Lampiran A – Berlawan dengan komputer

1. Pemain hanya perlu mengklik ikon ttt.html.



2. Kemudian, paparan permainan akan dikeluarkan seperti di bawah. Pemain boleh memulakan permainan dengan meletakkan nod pada kotak – kotak tersebut.



Main Semula

X menang : 0

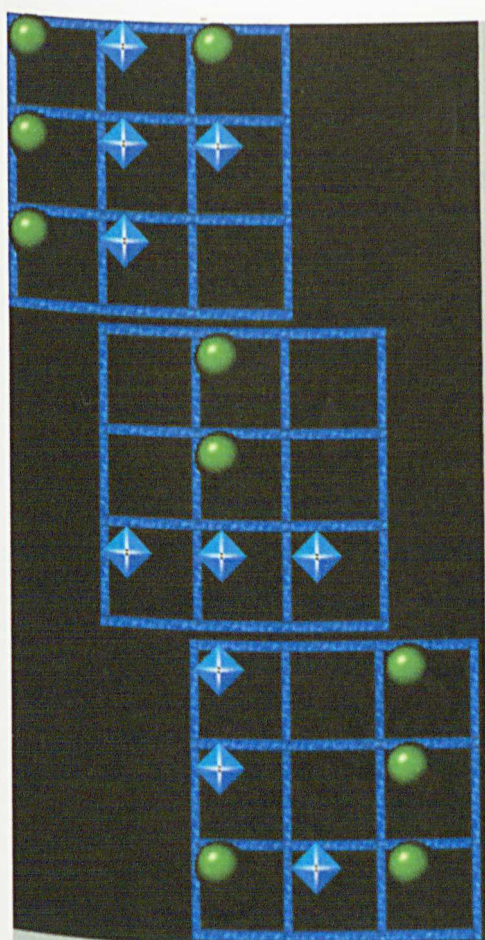
O menang : 0

Permainan ini adalah permainan 3D tic tac toe
Pemain boleh memenangi dengan meletakkan nod pada
satu baris secara menegak, melintang, pepenjuru
Pemain akan ditentukan sebagai pemenang mengikut
jumlah baris yang dimenangi

Semoga anda terhibur dengan permainan 3DTTT ini

Sekian, terima kasih

3. Paparan ketika permainan sedang dilaksanakan. Pemain yang mendapat barisan paling banyak dianggap sebagai pemenang. Pemain akan menggunakan imej X manakala komputer menggunakan imej O.



Main Semula

X menang : 3

O menang : 7

Permainan ini adalah permainan 3D tic tac toe
Pemain boleh memenangi dengan meletakkan nod pada
satu baris secara menegak, melintang, pepenjuru
Pemain akan ditentukan sebagai pemenang mengikut
jumlah baris yang dimenangi

Semoga anda terhibur dengan permainan 3DTTT ini

Sekian, terima kasih

4. Di sini disertakan kod program bagi permainan 3D Tic Tac Toe berlawan dengan komputer. Kod program ini boleh dilihat dengan menggunakan notepad ataupun perisian Java contoh JCreator.

```
import java.awt.*;
import java.applet.*;
import java.awt.image.*;

public class ttt extends Applet {

    Button newgame;
    Image backgrd;
    Image nodeO,nodeX;
    int boxes[] = new int [30];
    int winner1,winner2;
    boolean found,full,validresponse;
    String xwins,ywins;
    GridBagLayout thisLayout = new GridBagLayout();
    GridBagConstraints c = new GridBagConstraints ();

    public void init () {
        backgrd = getImage(getCodeBase(),"bg.jpg");
        c.fill = GridBagConstraints.BOTH;
        c.gridx=12;
        c.gridy=1;
        c.gridwidth=1;
        c.gridheight=1;
        Button newButton = new Button ( "Main Semula");
        thisLayout.setConstraints(newButton,c);
        add(newButton);

        xwins="X menang :";
        ywins="O menang :";
        reset ();

    } // init

    public boolean action( Event e, Object obj )
    {
        if ( "Main Semula".equals( obj ) ) {
            reset ();
        }
    }
}
```



```
return true;
}
```

```
public void reset () {
    xwins="X menang :";
    ywins="O menang :";
    for (int p=0;p<30;p++) boxes[p]=0;
    winner1=0;
    winner2=0;
    repaint ();
}
```

```
public void paint (Graphics g) {
    g.drawImage (backgrd,0,0,this);
    int p;
    for (p=0;p<27;p++) if (boxes[p]==1) drawX (g,p);
    for (p=0;p<27;p++) if (boxes[p]==2) drawO (g,p);
    g.setColor (Color.black);
    g.drawString (xwins,280,40);
    g.drawString (ywins,280,60);
    g.drawString ("Permainan ini adalah permainan 3D tic tac
        toe",280,180);
    g.drawString ("Pemain boleh memenangi dengan meletakkan nod
        pada",280,200);
    g.drawString ("satu baris secara menegak, melintang,
        pepenjuru",280,220);
    g.drawString ("Pemain akan ditentukan sebagai pemenang
        mengikut",280,240);
    g.drawString ("jumlah baris yang dimenangi",280,260);
    g.drawString ("Semoga anda terhibur dengan permainan 3DTTT
        ini",280,340);
    g.drawString ("Sekian, terima kasih",280,380);
}
```

```
public void drawX (Graphics g,int where) {
    nodeX = getImage(getCodeBase(),"ex.gif");
    int x,y,xo1,xo2,yo1,yo2;
    xo1=50;
    yo1=9;
    xo2=98;
    yo2=17;
    g.setColor (Color.white);
    x=(where-((int) (where/3)*3)) * 50;
    y=((int) (where/3))*50;
    if (where<9) {
        g.drawImage(nodeX,x,y,this);
    }
}
```

```

        if (where>8) if (where<18) {
            g.drawImage(nodeX,x+xo1,y+yo1,this);
        }
        if (where>17) if (where<27) {
            g.drawImage(nodeX,x+xo2,y+yo2,this);
        }
    }

    public void drawO (Graphics g,int where) {
        nodeO = getImage(getCodeBase(),"ball.gif");
        int x,y,xo1,xo2,yo1,yo2;
        xo1=50;
        yo1=9;
        xo2=98;
        yo2=17;
        //g.setColor (Color.white);
        x=(where-((int) (where/3)*3)) * 50;
        y=((int) (where/3))*50;
        if (where<9)
            g.drawImage(nodeO,x,y,this);

        if (where>8) if (where<18)
            g.drawImage(nodeO,x+xo1,y+yo1,this);

        if (where>17) if (where<27)
            g.drawImage(nodeO,x+xo2,y+yo2,this);
    }

    public boolean mouseDown( Event e, int x, int y )
    {
        int p,px,py,xo1,xo2,yo1,yo2;
        xo1=50;
        yo1=9;
        xo2=98;
        yo2=17;
        validresponse=false;

        for (p=0;p<9;p++) {
            px=p-((int) (p/3)*3);
            py=(int) (p/3);
            if (x>px*50) if (x<(px*50)+50) if (y>py*50) if (y<(py*50)+50)
            if (boxes[p]==0) {
                boxes[p]=1;
                validresponse=true;
            }
        }
    }

```

```

for (p=9;p<18;p++) {
    px=p-((int) (p/3)*3);
    py=(int) (p/3);
    if (x>px*50+xo1) if (x<(px*50)+50+xo1) if (y>py*50+yo1) if
(y<(py*50)+50+yo1)
        if (boxes[p]==0) {
            validresponse=true;
            boxes[p]=1;
        }
    }

```

```

for (p=18;p<27;p++) {
    px=p-((int) (p/3)*3);
    py=(int) (p/3);
    if (x>px*50+xo2) if (x<(px*50)+50+xo2) if (y>py*50+yo2) if
(y<(py*50)+50+yo2)
        if (boxes[p]==0) {
            validresponse=true;
            boxes[p]=1;
        }
    }

```

```

if (validresponse==true) computermove ();

```

```

checkwin ();

```

```

repaint ();
    return true;
}

```

```

public void checkfull () {
    full=true;
    for (int p=0;p<27;p++) if (boxes[p]==0) full=false;
}

```

```

public void computermove () {
    found=false;
    if (found==false) goforcenter ();
    if (found==false) goforwin ();
    if (found==false) goforblock ();
    if (found==false) putrandom ();
}

```

```

public void goforcenter () {
    if (boxes[13]==0) {
        boxes[13]=2;
    }
}

```



```

        found=true;
    }
}

//algoritma permainan
public void goforwin () {
    int p,three,bp,gotone;
    gotone=0;
    int first = 2;

    bp=0;
    three=0;
    for (p=0;p<3;p=p+1) if (boxes [p]==first) three=three+1;
    if (three==2) for (p=0;p<3;p=p+1) if (boxes [p]==0) {
        bp=p;
        gotone=1;
    }
    three=0;
    for (p=3;p<6;p=p+1) if (boxes [p]==first) three=three+1;
    if (three==2) for (p=3;p<6;p=p+1) if (boxes [p]==0) {
        bp=p;
        gotone=1;
    }
    three=0;
    for (p=6;p<9;p=p+1) if (boxes [p]==first) three=three+1;
    if (three==2) for (p=6;p<9;p=p+1) if (boxes [p]==0) {
        bp=p;
        gotone=1;
    }
    three=0;
    for (p=0;p<7;p=p+3) if (boxes [p]==first) three=three+1;
    if (three==2) for (p=0;p<7;p=p+3) if (boxes [p]==0) {
        bp=p;
        gotone=1;
    }
    three=0;
    for (p=1;p<8;p=p+3) if (boxes [p]==first) three=three+1;
    if (three==2) for (p=1;p<8;p=p+3) if (boxes [p]==0) {
        bp=p;
        gotone=1;
    }
    three=0;
    for (p=2;p<9;p=p+3) if (boxes [p]==first) three=three+1;
    if (three==2) for (p=2;p<9;p=p+3) if (boxes [p]==0) {
        bp=p;
        gotone=1;
    }
}

```

```

three=0;
for (p=0;p<9;p=p+4) if (boxes [p]==first) three=three+1;
if (three==2) for (p=0;p<9;p=p+4) if (boxes [p]==0) {
bp=p;
gotone=1;
}
three=0;
for (p=2;p<7;p=p+2) if (boxes [p]==first) three=three+1;
if (three==3) for (p=2;p<7;p=p+2) if (boxes [p]==0) {
bp=p;
gotone=1;
}
three=0;
three=0;
for (p=9;p<12;p=p+1) if (boxes [p]==first) three=three+1;
if (three==2) for (p=9;p<12;p=p+1) if (boxes [p]==0) {
bp=p;
gotone=1;
}
three=0;
for (p=12;p<15;p=p+1) if (boxes [p]==first) three=three+1;
if (three==2) for (p=12;p<15;p=p+1) if (boxes [p]==0) {
bp=p;
gotone=1;
}
three=0;
for (p=15;p<18;p=p+1) if (boxes [p]==first) three=three+1;
if (three==2) for (p=15;p<18;p=p+1) if (boxes [p]==0) {
bp=p;
gotone=1;
}
three=0;
for (p=9;p<16;p=p+3) if (boxes [p]==first) three=three+1;
if (three==2) for (p=9;p<16;p=p+3) if (boxes [p]==0) {
bp=p;
gotone=1;
}
three=0;
for (p=10;p<17;p=p+3) if (boxes [p]==first) three=three+1;
if (three==2) for (p=10;p<17;p=p+3) if (boxes [p]==0) {
bp=p;
gotone=1;
}
three=0;
for (p=11;p<18;p=p+3) if (boxes [p]==first) three=three+1;
if (three==2) for (p=11;p<18;p=p+3) if (boxes [p]==0) {

```

```

bp=p;
gotone=1;
}
three=0;
for (p=9;p<18;p=p+4) if (boxes [p]==first) three=three+1;
if (three==2) for (p=9;p<18;p=p+4) if (boxes [p]==0) {
bp=p;
gotone=1;
}
three=0;
for (p=11;p<16;p=p+2) if (boxes [p]==first) three=three+1;
if (three==3) for (p=11;p<16;p=p+2) if (boxes [p]==0) {
bp=p;
gotone=1;
}
three=0;
three=0;
for (p=18;p<21;p=p+1) if (boxes [p]==first) three=three+1;
if (three==2) for (p=18;p<21;p=p+1) if (boxes [p]==0) {
bp=p;
gotone=1;
}
three=0;
for (p=21;p<24;p=p+1) if (boxes [p]==first) three=three+1;
if (three==2) for (p=21;p<24;p=p+1) if (boxes [p]==0) {
bp=p;
gotone=1;
}
three=0;
for (p=24;p<27;p=p+1) if (boxes [p]==first) three=three+1;
if (three==2) for (p=24;p<27;p=p+1) if (boxes [p]==0) {
bp=p;
gotone=1;
}
three=0;
for (p=18;p<25;p=p+3) if (boxes [p]==first) three=three+1;
if (three==2) for (p=18;p<25;p=p+3) if (boxes [p]==0) {
bp=p;
gotone=1;
}
three=0;
for (p=19;p<26;p=p+3) if (boxes [p]==first) three=three+1;
if (three==2) for (p=19;p<26;p=p+3) if (boxes [p]==0) {
bp=p;
gotone=1;
}

```



```

three=0;
for (p=20;p<27;p=p+3) if (boxes [p]==first) three=three+1;
if (three==2) for (p=20;p<27;p=p+3) if (boxes [p]==0) {
bp=p;
gotone=1;
}
three=0;
for (p=18;p<27;p=p+4) if (boxes [p]==first) three=three+1;
if (three==2) for (p=18;p<27;p=p+4) if (boxes [p]==0) {
bp=p;
gotone=1;
}
three=0;
for (p=20;p<25;p=p+2) if (boxes [p]==first) three=three+1;
if (three==2) for (p=20;p<25;p=p+2) if (boxes [p]==0) {
bp=p;
gotone=1;
}
three=0;
for (p=0;p<19;p=p+9) if (boxes [p]==first) three=three+1;
if (three==2) for (p=0;p<19;p=p+9) if (boxes [p]==0) {
bp=p;
gotone=1;
}
three=0;
for (p=1;p<20;p=p+9) if (boxes [p]==first) three=three+1;
if (three==2) for (p=1;p<20;p=p+9) if (boxes [p]==0) {
bp=p;
gotone=1;
}
three=0;
for (p=2;p<21;p=p+9) if (boxes [p]==first) three=three+1;
if (three==2) for (p=2;p<21;p=p+9) if (boxes [p]==0) {
bp=p;
gotone=1;
}
three=0;
for (p=3;p<22;p=p+9) if (boxes [p]==first) three=three+1;
if (three==2) for (p=3;p<22;p=p+9) if (boxes [p]==0) {
bp=p;
gotone=1;
}
three=0;
for (p=4;p<23;p=p+9) if (boxes [p]==first) three=three+1;
if (three==2) for (p=4;p<23;p=p+9) if (boxes [p]==0) {
bp=p;

```

```

gotone=1;
}
three=0;
for (p=5;p<24;p=p+9) if (boxes [p]==first) three=three+1;
if (three==2) for (p=5;p<24;p=p+9) if (boxes [p]==0) {
bp=p;
gotone=1;
}
three=0;
for (p=6;p<25;p=p+9) if (boxes [p]==first) three=three+1;
if (three==2) for (p=6;p<25;p=p+9) if (boxes [p]==0) {
bp=p;
gotone=1;
}
three=0;
for (p=7;p<26;p=p+9) if (boxes [p]==first) three=three+1;
if (three==2) for (p=7;p<26;p=p+9) if (boxes [p]==0) {
bp=p;
gotone=1;
}
three=0;
for (p=8;p<27;p=p+9) if (boxes [p]==first) three=three+1;
if (three==2) for (p=8;p<27;p=p+9) if (boxes [p]==0) {
bp=p;
gotone=1;
}
three=0;
for (p=0;p<25;p=p+12) if (boxes [p]==first) three=three+1;
if (three==2) for (p=0;p<25;p=p+12) if (boxes [p]==0) {
bp=p;
gotone=1;
}
three=0;
for (p=1;p<26;p=p+12) if (boxes [p]==first) three=three+1;
if (three==2) for (p=1;p<26;p=p+12) if (boxes [p]==0) {
bp=p;
gotone=1;
}
three=0;
for (p=2;p<27;p=p+12) if (boxes [p]==first) three=three+1;
if (three==2) for (p=2;p<27;p=p+12) if (boxes [p]==0) {
bp=p;
gotone=1;
}
three=0;
for (p=6;p<19;p=p+6) if (boxes [p]==first) three=three+1;

```

```

if (three==2) for (p=6;p<19;p=p+6) if (boxes [p]==0) {
bp=p;
gotone=1;
}
three=0;
for (p=7;p<20;p=p+6) if (boxes [p]==first) three=three+1;
if (three==2) for (p=7;p<20;p=p+6) if (boxes [p]==0) {
bp=p;
gotone=1;
}
three=0;
for (p=8;p<21;p=p+6) if (boxes [p]==first) three=three+1;
if (three==2) for (p=8;p<21;p=p+6) if (boxes [p]==0) {
bp=p;
gotone=1;
}
three=0;
for (p=0;p<21;p=p+10) if (boxes [p]==first) three=three+1;
if (three==2) for (p=0;p<21;p=p+10) if (boxes [p]==0) {
bp=p;
gotone=1;
}
three=0;
for (p=3;p<24;p=p+10) if (boxes [p]==first) three=three+1;
if (three==2) for (p=3;p<24;p=p+10) if (boxes [p]==0) {
bp=p;
gotone=1;
}
three=0;
for (p=6;p<27;p=p+10) if (boxes [p]==first) three=three+1;
if (three==2) for (p=6;p<27;p=p+10) if (boxes [p]==0) {
bp=p;
gotone=1;
}
three=0;
for (p=2;p<19;p=p+8) if (boxes [p]==first) three=three+1;
if (three==2) for (p=2;p<19;p=p+8) if (boxes [p]==0) {
bp=p;
gotone=1;
}
three=0;
for (p=5;p<22;p=p+8) if (boxes [p]==first) three=three+1;
if (three==2) for (p=5;p<22;p=p+8) if (boxes [p]==0) {
bp=p;
gotone=1;
}

```

```

three=0;
for (p=8;p<25;p=p+8) if (boxes [p]==first) three=three+1;
if (three==2) for (p=8;p<25;p=p+8) if (boxes [p]==0) {
    bp=p;
    gotone=1;
}
three=0;
for (p=0;p<27;p=p+13) if (boxes [p]==first) three=three+1;
if (three==2) for (p=0;p<27;p=p+13) if (boxes [p]==0) {
    bp=p;
    gotone=1;
}
three=0;
for (p=2;p<25;p=p+11) if (boxes [p]==first) three=three+1;
if (three==2) for (p=2;p<25;p=p+11) if (boxes [p]==0) {
    bp=p;
    gotone=1;
}
three=0;
for (p=6;p<21;p=p+7) if (boxes [p]==first) three=three+1;
if (three==2) for (p=6;p<21;p=p+7) if (boxes [p]==0) {
    bp=p;
    gotone=1;
}
three=0;
for (p=8;p<19;p=p+5) if (boxes [p]==first) three=three+1;
if (three==2) for (p=8;p<19;p=p+5) if (boxes [p]==0) {
    bp=p;
    gotone=1;
}
if (gotone==1) {
    found = true;
    boxes[bp]=2;
}
}

```

```

public void goforblock () {
    int p,three,ffirst,bp,gotone;
    ffirst=1;

```

```

    bp=0;
    gotone=0;
    three=0;

```

```

    for (p=0;p<3;p=p+1) if (boxes [p]==ffirst) three=three+1;
    if (three==2) for (p=0;p<3;p=p+1) if (boxes [p]==0) {

```



```

bp=p;
gotone=1;
}
three=0;
for (p=3;p<6;p=p+1) if (boxes [p]==ffirst) three=three+1;
if (three==2) for (p=3;p<6;p=p+1) if (boxes [p]==0) {
bp=p;
gotone=1;
}
three=0;
for (p=6;p<9;p=p+1) if (boxes [p]==ffirst) three=three+1;
if (three==2) for (p=6;p<9;p=p+1) if (boxes [p]==0) {
bp=p;
gotone=1;
}
three=0;
for (p=0;p<7;p=p+3) if (boxes [p]==ffirst) three=three+1;
if (three==2) for (p=0;p<7;p=p+3) if (boxes [p]==0) {
bp=p;
gotone=1;
}
three=0;
for (p=1;p<8;p=p+3) if (boxes [p]==ffirst) three=three+1;
if (three==2) for (p=1;p<8;p=p+3) if (boxes [p]==0) {
bp=p;
gotone=1;
}
three=0;
for (p=2;p<9;p=p+3) if (boxes [p]==ffirst) three=three+1;
if (three==2) for (p=2;p<9;p=p+3) if (boxes [p]==0) {
bp=p;
gotone=1;
}
three=0;
for (p=0;p<9;p=p+4) if (boxes [p]==ffirst) three=three+1;
if (three==2) for (p=0;p<9;p=p+4) if (boxes [p]==0) {
bp=p;
gotone=1;
}
three=0;
for (p=2;p<7;p=p+2) if (boxes [p]==ffirst) three=three+1;
if (three==3) for (p=2;p<7;p=p+2) if (boxes [p]==0) {
bp=p;
gotone=1;
}
three=0;

```

```

for (p=9;p<12;p=p+1) if (boxes [p]==ffirst) three=three+1;
if (three==2) for (p=9;p<12;p=p+1) if (boxes [p]==0) {
bp=p;
gotone=1;
}
three=0;
for (p=12;p<15;p=p+1) if (boxes [p]==ffirst) three=three+1;
if (three==2) for (p=12;p<15;p=p+1) if (boxes [p]==0) {
bp=p;
gotone=1;
}
three=0;
for (p=15;p<18;p=p+1) if (boxes [p]==ffirst) three=three+1;
if (three==2) for (p=15;p<18;p=p+1) if (boxes [p]==0) {
bp=p;
gotone=1;
}
three=0;
for (p=9;p<16;p=p+3) if (boxes [p]==ffirst) three=three+1;
if (three==2) for (p=9;p<16;p=p+3) if (boxes [p]==0) {
bp=p;
gotone=1;
}
three=0;
for (p=10;p<17;p=p+3) if (boxes [p]==ffirst) three=three+1;
if (three==2) for (p=10;p<17;p=p+3) if (boxes [p]==0) {
bp=p;
gotone=1;
}
three=0;
for (p=11;p<18;p=p+3) if (boxes [p]==ffirst) three=three+1;
if (three==2) for (p=11;p<18;p=p+3) if (boxes [p]==0) {
bp=p;
gotone=1;
}
three=0;
for (p=9;p<18;p=p+4) if (boxes [p]==ffirst) three=three+1;
if (three==2) for (p=9;p<18;p=p+4) if (boxes [p]==0) {
bp=p;
gotone=1;
}
three=0;
for (p=11;p<16;p=p+2) if (boxes [p]==ffirst) three=three+1;
if (three==3) for (p=11;p<16;p=p+2) if (boxes [p]==0) {
bp=p;
gotone=1;
}

```

```

}
three=0;
three=0;
for (p=18;p<21;p=p+1) if (boxes [p]==ffirst) three=three+1;
if (three==2) for (p=18;p<21;p=p+1) if (boxes [p]==0) {
bp=p;
gotone=1;
}
three=0;
for (p=21;p<24;p=p+1) if (boxes [p]==ffirst) three=three+1;
if (three==2) for (p=21;p<24;p=p+1) if (boxes [p]==0) {
bp=p;
gotone=1;
}
three=0;
for (p=24;p<27;p=p+1) if (boxes [p]==ffirst) three=three+1;
if (three==2) for (p=24;p<27;p=p+1) if (boxes [p]==0) {
bp=p;
gotone=1;
}
three=0;
for (p=18;p<25;p=p+3) if (boxes [p]==ffirst) three=three+1;
if (three==2) for (p=18;p<25;p=p+3) if (boxes [p]==0) {
bp=p;
gotone=1;
}
three=0;
for (p=19;p<26;p=p+3) if (boxes [p]==ffirst) three=three+1;
if (three==2) for (p=19;p<26;p=p+3) if (boxes [p]==0) {
bp=p;
gotone=1;
}
three=0;
for (p=20;p<27;p=p+3) if (boxes [p]==ffirst) three=three+1;
if (three==2) for (p=20;p<27;p=p+3) if (boxes [p]==0) {
bp=p;
gotone=1;
}
three=0;
for (p=18;p<27;p=p+4) if (boxes [p]==ffirst) three=three+1;
if (three==2) for (p=18;p<27;p=p+4) if (boxes [p]==0) {
bp=p;
gotone=1;
}
three=0;
for (p=20;p<25;p=p+2) if (boxes [p]==ffirst) three=three+1;

```

```

if (three==2) for (p=20;p<25;p=p+2) if (boxes [p]==0) {
    bp=p;
    gotone=1;
}
three=0;

```

```

if (gotone==1) {
    found = true;
    boxes[bp]=2;
}

```

```

}
public void putrandom () {
    int a,b;
    while (found==false) {
        a=(int) (26 * Math.random());
        if (boxes[a]==0) {
            found=true;
            boxes[a]=2;
        } //if
        checkfull ();
        if (full==true) found=true;
    } // while
} // putrand

```

```

void checkwin ()
{
    winner1=0;
    winner2=0;
    int p,three;
    three=0;
    for (p=0;p<3;p=p+1) if (boxes [p]==1) three=three+1;
    if (three==3) winner1=winner1+1;
    three=0;
    for (p=3;p<6;p=p+1) if (boxes [p]==1) three=three+1;
    if (three==3) winner1=winner1+1;
    three=0;
    for (p=6;p<9;p=p+1) if (boxes [p]==1) three=three+1;
    if (three==3) winner1=winner1+1;
    three=0;
    for (p=0;p<7;p=p+3) if (boxes [p]==1) three=three+1;
    if (three==3) winner1=winner1+1;
    three=0;
    for (p=1;p<8;p=p+3) if (boxes [p]==1) three=three+1;
    if (three==3) winner1=winner1+1;
    three=0;
    for (p=2;p<9;p=p+3) if (boxes [p]==1) three=three+1;
}

```



```

if (three==3) winner1=winner1+1;
three=0;
for (p=0;p<9;p=p+4) if (boxes [p]==1) three=three+1;
if (three==3) winner1=winner1+1;
three=0;
for (p=2;p<7;p=p+2) if (boxes [p]==1) three=three+1;
if (three==3) winner1=winner1+1;
three=0;
for (p=0;p<3;p=p+1) if (boxes [p]==2) three=three+1;
if (three==3) winner2=winner2+1;
three=0;
for (p=3;p<6;p=p+1) if (boxes [p]==2) three=three+1;
if (three==3) winner2=winner2+1;
three=0;
for (p=6;p<9;p=p+1) if (boxes [p]==2) three=three+1;
if (three==3) winner2=winner2+1;
three=0;
for (p=0;p<7;p=p+3) if (boxes [p]==2) three=three+1;
if (three==3) winner2=winner2+1;
three=0;
for (p=1;p<8;p=p+3) if (boxes [p]==2) three=three+1;
if (three==3) winner2=winner2+1;
three=0;
for (p=2;p<9;p=p+3) if (boxes [p]==2) three=three+1;
if (three==3) winner2=winner2+1;
three=0;
for (p=0;p<9;p=p+4) if (boxes [p]==2) three=three+1;
if (three==3) winner2=winner2+1;
three=0;
for (p=2;p<7;p=p+2) if (boxes [p]==2) three=three+1;
if (three==3) winner2=winner2+1;
three=0;
for (p=9;p<12;p=p+1) if (boxes [p]==1) three=three+1;
if (three==3) winner1=winner1+1;
three=0;
for (p=12;p<15;p=p+1) if (boxes [p]==1) three=three+1;
if (three==3) winner1=winner1+1;
three=0;
for (p=15;p<18;p=p+1) if (boxes [p]==1) three=three+1;
if (three==3) winner1=winner1+1;
three=0;
for (p=9;p<16;p=p+3) if (boxes [p]==1) three=three+1;
if (three==3) winner1=winner1+1;
three=0;
for (p=10;p<17;p=p+3) if (boxes [p]==1) three=three+1;
if (three==3) winner1=winner1+1;

```

```

three=0;
for (p=11;p<18;p=p+3) if (boxes [p]==1) three=three+1;
if (three==3) winner1=winner1+1;
three=0;
for (p=9;p<18;p=p+4) if (boxes [p]==1) three=three+1;
if (three==3) winner1=winner1+1;
three=0;
for (p=11;p<16;p=p+2) if (boxes [p]==1) three=three+1;
if (three==3) winner1=winner1+1;
three=0;
for (p=9;p<12;p=p+1) if (boxes [p]==2) three=three+1;
if (three==3) winner2=winner2+1;
three=0;
for (p=12;p<15;p=p+1) if (boxes [p]==2) three=three+1;
if (three==3) winner2=winner2+1;
three=0;
for (p=15;p<18;p=p+1) if (boxes [p]==2) three=three+1;
if (three==3) winner2=winner2+1;
three=0;
for (p=9;p<16;p=p+3) if (boxes [p]==2) three=three+1;
if (three==3) winner2=winner2+1;
three=0;
for (p=10;p<17;p=p+3) if (boxes [p]==2) three=three+1;
if (three==3) winner2=winner2+1;
three=0;
for (p=11;p<18;p=p+3) if (boxes [p]==2) three=three+1;
if (three==3) winner2=winner2+1;
three=0;
for (p=9;p<18;p=p+4) if (boxes [p]==2) three=three+1;
if (three==3) winner2=winner2+1;
three=0;
for (p=11;p<16;p=p+2) if (boxes [p]==2) three=three+1;
if (three==3) winner2=winner2+1;
three=0;
for (p=18;p<21;p=p+1) if (boxes [p]==1) three=three+1;
if (three==3) winner1=winner1+1;
three=0;
for (p=21;p<24;p=p+1) if (boxes [p]==1) three=three+1;
if (three==3) winner1=winner1+1;
three=0;
for (p=24;p<27;p=p+1) if (boxes [p]==1) three=three+1;
if (three==3) winner1=winner1+1;
three=0;
for (p=18;p<25;p=p+3) if (boxes [p]==1) three=three+1;
if (three==3) winner1=winner1+1;
three=0;

```

```

for (p=19;p<26;p=p+3) if (boxes [p]==1) three=three+1;
if (three==3) winner1=winner1+1;
three=0;
for (p=20;p<27;p=p+3) if (boxes [p]==1) three=three+1;
if (three==3) winner1=winner1+1;
three=0;
for (p=18;p<27;p=p+4) if (boxes [p]==1) three=three+1;
if (three==3) winner1=winner1+1;
three=0;
for (p=20;p<25;p=p+2) if (boxes [p]==1) three=three+1;
if (three==3) winner1=winner1+1;
three=0;
for (p=18;p<21;p=p+1) if (boxes [p]==2) three=three+1;
if (three==3) winner2=winner2+1;
three=0;
for (p=21;p<24;p=p+1) if (boxes [p]==2) three=three+1;
if (three==3) winner2=winner2+1;
three=0;
for (p=24;p<27;p=p+1) if (boxes [p]==2) three=three+1;
if (three==3) winner2=winner2+1;
three=0;
for (p=18;p<25;p=p+3) if (boxes [p]==2) three=three+1;
if (three==3) winner2=winner2+1;
three=0;
for (p=19;p<26;p=p+3) if (boxes [p]==2) three=three+1;
if (three==3) winner2=winner2+1;
three=0;
for (p=20;p<27;p=p+3) if (boxes [p]==2) three=three+1;
if (three==3) winner2=winner2+1;
three=0;
for (p=18;p<27;p=p+4) if (boxes [p]==2) three=three+1;
if (three==3) winner2=winner2+1;
three=0;
for (p=20;p<25;p=p+2) if (boxes [p]==2) three=three+1;
if (three==3) winner2=winner2+1;
three=0;
for (p=0;p<19;p=p+9) if (boxes[p]==1) three=three+1;
if (three==3) winner1=winner1+1;
three=0;
for (p=1;p<20;p=p+9) if (boxes[p]==1) three=three+1;
if (three==3) winner1=winner1+1;
three=0;
for (p=2;p<21;p=p+9) if (boxes[p]==1) three=three+1;
if (three==3) winner1=winner1+1;
three=0;
for (p=3;p<22;p=p+9) if (boxes[p]==1) three=three+1;

```



```

if (three==3) winner1=winner1+1;
three=0;
for (p=4;p<23;p=p+9) if (boxes[p]==1) three=three+1;
if (three==3) winner1=winner1+1;
three=0;
for (p=5;p<24;p=p+9) if (boxes[p]==1) three=three+1;
if (three==3) winner1=winner1+1;
three=0;
for (p=6;p<25;p=p+9) if (boxes[p]==1) three=three+1;
if (three==3) winner1=winner1+1;
three=0;
for (p=7;p<26;p=p+9) if (boxes[p]==1) three=three+1;
if (three==3) winner1=winner1+1;
three=0;
for (p=8;p<27;p=p+9) if (boxes[p]==1) three=three+1;
if (three==3) winner1=winner1+1;
three=0;
for (p=0;p<25;p=p+12) if (boxes[p]==1) three=three+1;
if (three==3) winner1=winner1+1;
three=0;
for (p=1;p<26;p=p+12) if (boxes[p]==1) three=three+1;
if (three==3) winner1=winner1+1;
three=0;
for (p=2;p<27;p=p+12) if (boxes[p]==1) three=three+1;
if (three==3) winner1=winner1+1;
three=0;
for (p=6;p<19;p=p+6) if (boxes[p]==1) three=three+1;
if (three==3) winner1=winner1+1;
three=0;
for (p=7;p<20;p=p+6) if (boxes[p]==1) three=three+1;
if (three==3) winner1=winner1+1;
three=0;
for (p=8;p<21;p=p+6) if (boxes[p]==1) three=three+1;
if (three==3) winner1=winner1+1;
three=0;
for (p=0;p<21;p=p+10) if (boxes[p]==1) three=three+1;
if (three==3) winner1=winner1+1;
three=0;
for (p=3;p<24;p=p+10) if (boxes[p]==1) three=three+1;
if (three==3) winner1=winner1+1;
three=0;
for (p=6;p<27;p=p+10) if (boxes[p]==1) three=three+1;
if (three==3) winner1=winner1+1;
three=0;
for (p=2;p<19;p=p+8) if (boxes[p]==1) three=three+1;
if (three==3) winner1=winner1+1;

```



```

three=0;
for (p=5;p<22;p=p+8) if (boxes[p]==1) three=three+1;
if (three==3) winner1=winner1+1;
three=0;
for (p=8;p<25;p=p+8) if (boxes[p]==1) three=three+1;
if (three==3) winner1=winner1+1;
three=0;
for (p=0;p<27;p=p+13) if (boxes[p]==1) three=three+1;
if (three==3) winner1=winner1+1;
three=0;
for (p=2;p<25;p=p+11) if (boxes[p]==1) three=three+1;
if (three==3) winner1=winner1+1;
three=0;
for (p=6;p<21;p=p+7) if (boxes[p]==1) three=three+1;
if (three==3) winner1=winner1+1;
three=0;
for (p=8;p<19;p=p+5) if (boxes[p]==1) three=three+1;
if (three==3) winner1=winner1+1;
three=0;
for (p=0;p<19;p=p+9) if (boxes[p]==2) three=three+1;
if (three==3) winner2=winner2+1;
three=0;
for (p=1;p<20;p=p+9) if (boxes[p]==2) three=three+1;
if (three==3) winner2=winner2+1;
three=0;
for (p=2;p<21;p=p+9) if (boxes[p]==2) three=three+1;
if (three==3) winner2=winner2+1;
three=0;
for (p=3;p<22;p=p+9) if (boxes[p]==2) three=three+1;
if (three==3) winner2=winner2+1;
three=0;
for (p=4;p<23;p=p+9) if (boxes[p]==2) three=three+1;
if (three==3) winner2=winner2+1;
three=0;
for (p=5;p<24;p=p+9) if (boxes[p]==2) three=three+1;
if (three==3) winner2=winner2+1;
three=0;
for (p=6;p<25;p=p+9) if (boxes[p]==2) three=three+1;
if (three==3) winner2=winner2+1;
three=0;
for (p=7;p<26;p=p+9) if (boxes[p]==2) three=three+1;
if (three==3) winner2=winner2+1;
three=0;
for (p=8;p<27;p=p+9) if (boxes[p]==2) three=three+1;
if (three==3) winner2=winner2+1;
three=0;

```

```

for (p=0;p<25;p=p+12) if (boxes[p]==2) three=three+1;
if (three==3) winner2=winner2+1;
three=0;
for (p=1;p<26;p=p+12) if (boxes[p]==2) three=three+1;
if (three==3) winner2=winner2+1;
three=0;
for (p=2;p<27;p=p+12) if (boxes[p]==2) three=three+1;
if (three==3) winner2=winner2+1;
three=0;
for (p=6;p<19;p=p+6) if (boxes[p]==2) three=three+1;
if (three==3) winner2=winner2+1;
three=0;
for (p=7;p<20;p=p+6) if (boxes[p]==2) three=three+1;
if (three==3) winner2=winner2+1;
three=0;
for (p=8;p<21;p=p+6) if (boxes[p]==2) three=three+1;
if (three==3) winner2=winner2+1;
three=0;
for (p=0;p<21;p=p+10) if (boxes[p]==2) three=three+1;
if (three==3) winner2=winner2+1;
three=0;
for (p=3;p<24;p=p+10) if (boxes[p]==2) three=three+1;
if (three==3) winner2=winner2+1;
three=0;
for (p=6;p<27;p=p+10) if (boxes[p]==2) three=three+1;
if (three==3) winner2=winner2+1;
three=0;
for (p=2;p<19;p=p+8) if (boxes[p]==2) three=three+1;
if (three==3) winner2=winner2+1;
three=0;
for (p=5;p<22;p=p+8) if (boxes[p]==2) three=three+1;
if (three==3) winner2=winner2+1;
three=0;
for (p=8;p<25;p=p+8) if (boxes[p]==2) three=three+1;
if (three==3) winner2=winner2+1;
three=0;
for (p=0;p<27;p=p+13) if (boxes[p]==2) three=three+1;
if (three==3) winner2=winner2+1;
three=0;
for (p=2;p<25;p=p+11) if (boxes[p]==2) three=three+1;
if (three==3) winner2=winner2+1;
three=0;
for (p=6;p<21;p=p+7) if (boxes[p]==2) three=three+1;
if (three==3) winner2=winner2+1;
three=0;
for (p=8;p<19;p=p+5) if (boxes[p]==2) three=three+1;

```



```

if (three==3) winner2=winner2+1;
three=0;
xwins="X menang : "+winner1;
ywins="O menang : "+winner2;
}

} // class TTT

```

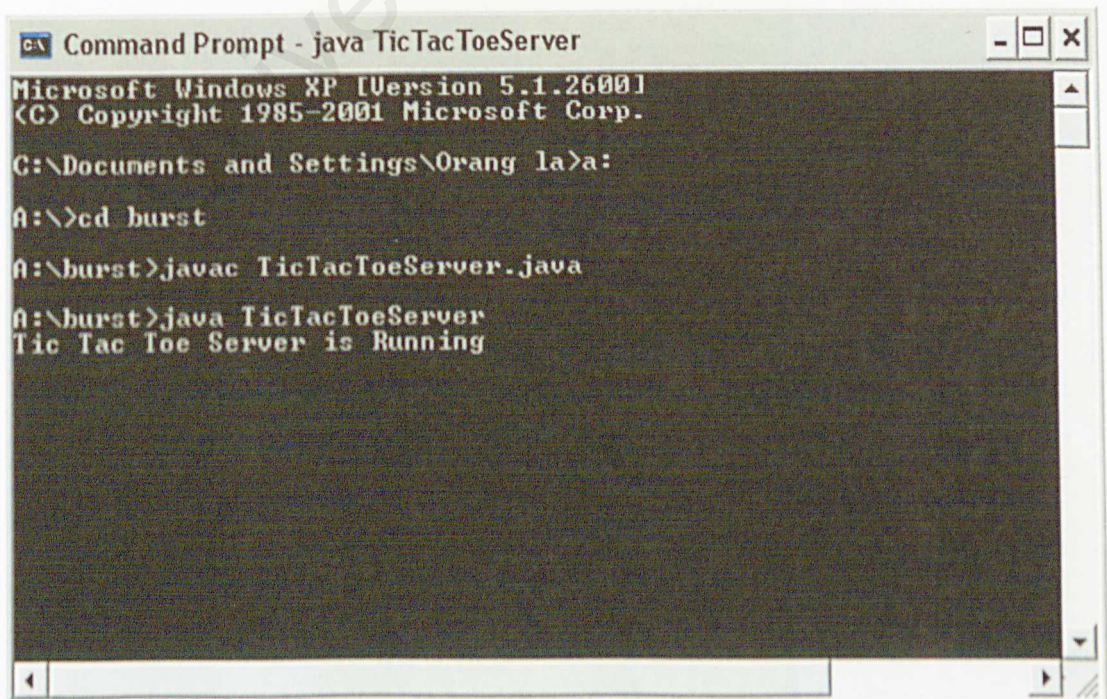
Lampiran B – Berlawan dengan pengguna yang lain (rangkaian)

1. Pemain perlu membuka pelayan permainan dengan menggunakan Command Prompt. Kemudian laksanakan arahan seperti di bawah. Pelayan permainan dilarikan dan permainan boleh dimulakan dengan menggunakan bahagian pelanggan.

```

Kompil      :   javac TicTacToeServer.java
Larian      :   java TicTacToeServer

```



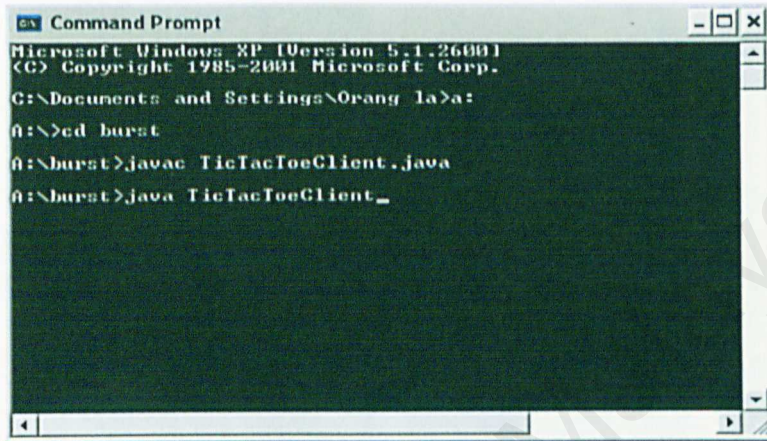
```

C:\> Command Prompt - java TicTacToeServer
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.
C:\Documents and Settings\Orang la>a:
A:\>cd burst
A:\burst>javac TicTacToeServer.java
A:\burst>java TicTacToeServer
Tic Tac Toe Server is Running

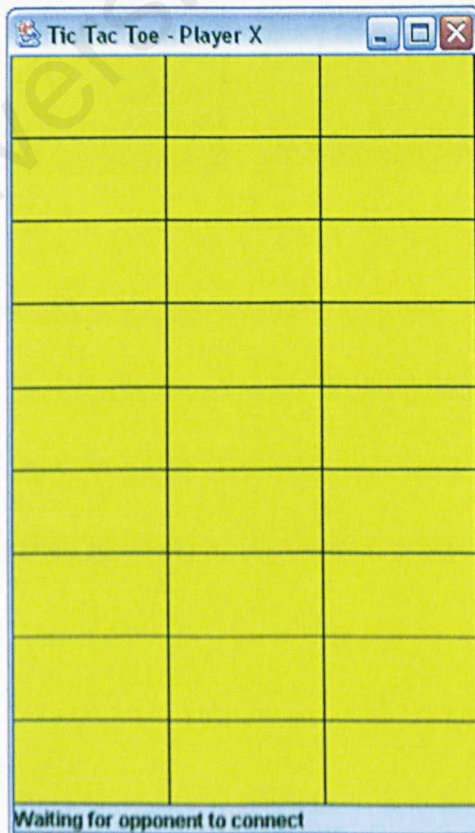
```


2. Bahagian pelanggan permainan dimulakan dengan menggunakan Command Prompt. Pemain perlu mengikut arahan seperti di bawah.

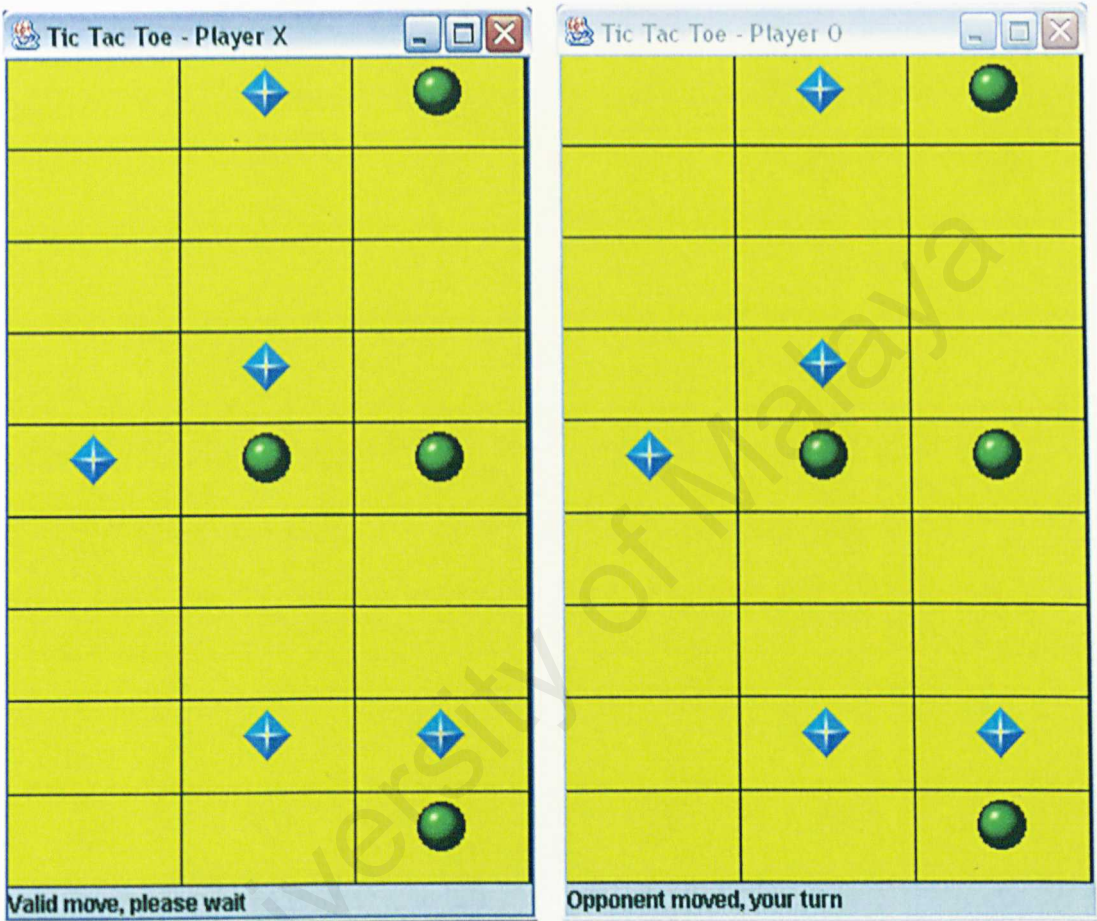
Kompil : javac TicTacToeClient.java
Larian : java TicTacToeClient



3. Paparan dibawah akan terhasil, pemain pertama perlu menunggu pemain lain dihubungkan untuk bermain antara satu sama lain.

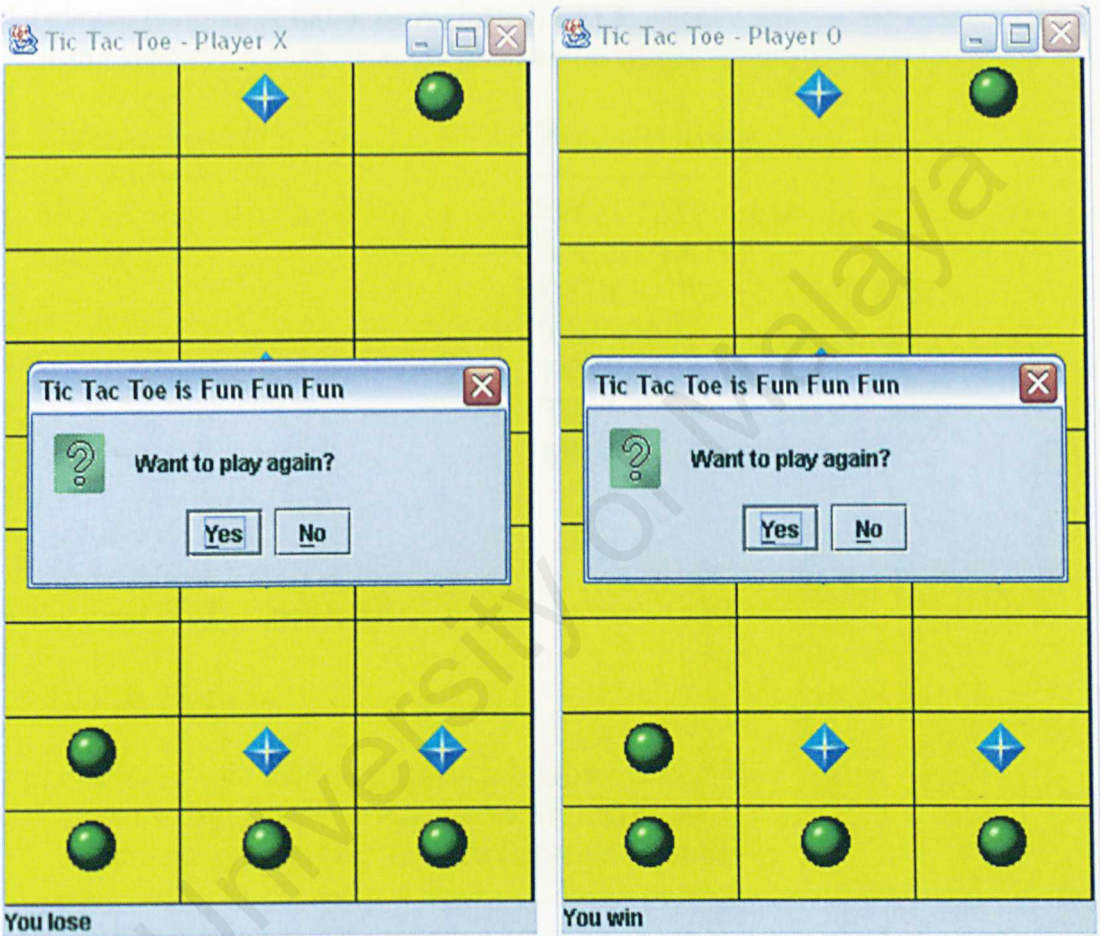


4. Setelah pemain kedua dihubungkan, permainan boleh dimulakan. Paparan ketika permainan sedang berjalan.



Pemain akan bermain mengikut giliran masing – masing, “Valid move, please wait” dan juga “Opponent moved, your turn” jelas menunjukkan giliran setiap pemain. Oleh itu, pemain tidak dapat berjalan jika bukan gilirannya ketika itu. Jika pemain meletakkan nod ketika bukan gilirannya, akan keluar amaran “ ? ” .

5. Apabila antara pemain telah menang, akan keluar kotak pilihan samada ingin meneruskan permainan ataupun tidak. Status pemain juga akan dipaparkan samada “You Win” ataupun “You Lose”.



6. Jika kedua – dua pemain memilih “Yes”, maka papan akan dikosongkan dan permainan dimulakan. Jika antara pemain memilih “No”, maka pemain perlu menunggu pemain lain dihubungkan.

7. Di sini disertakan kod program bagi pelayan permainan.

```
import java.net.Socket;
import java.net.ServerSocket;
import java.io.InputStreamReader;
import java.io.BufferedReader;
import java.io.PrintWriter;
import java.io.IOException;

/**
 * Pelayan permainan untuk permainan 3D TTT
 * String yang digunakan dalam permainan ini adalah seperti berikut:
 *
 * Client -> Server                Server -> Client
 * -----
 * MOVE <n> (0 <= n <= 26)         WELCOME <char> (char in {X, O})
 * QUIT                            VALID_MOVE
 *                                OTHER_PLAYER_MOVED <n>
 *                                VICTORY
 *                                DEFEAT
 *                                TIE
 *                                MESSAGE <text>
 */
```

```
public class TicTacToeServer {
```

```
    //aplikasi dilarikan
```

```
    public static void main(String[] args) throws Exception {
        ServerSocket listener = new ServerSocket(8901);
        System.out.println("Tic Tac Toe Server is Running");
        try {
            while (true) {
                Game game = new Game();
                Game.Player playerX = game.new Player(listener.accept(), 'X');
                Game.Player playerO = game.new Player(listener.accept(), 'O');
                playerX.setOpponent(playerO);
                playerO.setOpponent(playerX);
                game.currentPlayer = playerX;
                playerX.start();
                playerO.start();
            }
        } finally {
            listener.close();
        }
    }
}
```

```

    }
}

```

```

//permainan dua pemain
class Game {

```

```

    //papan mempunyai 27 kotak
    private Player[] board = {
        null, null, null,
        null, null, null,
        null, null, null,
        null, null, null,
        null, null, null,
        null, null, null,
        null, null, null,
        null, null, null,
        null, null, null};

```

```

    Player currentPlayer;

```

```

    //algoritma permainan
    public boolean hasWinner() {
        return

```

```

        (board[0] != null && board[0] == board[1] && board[0] == board[2])
        ||(board[3] != null && board[3] == board[4] && board[3] == board[5])
        ||(board[6] != null && board[6] == board[7] && board[6] == board[8])
        ||(board[0] != null && board[0] == board[3] && board[0] == board[6])
        ||(board[1] != null && board[1] == board[4] && board[1] == board[7])
        ||(board[2] != null && board[2] == board[5] && board[2] == board[8])
        ||(board[0] != null && board[0] == board[4] && board[0] == board[8])
        ||(board[2] != null && board[2] == board[4] && board[2] == board[6])
        ||(board[9] != null && board[9] == board[10] && board[9] == board[11])
        ||(board[12] != null && board[12] == board[13] && board[12] == board[14])
        ||(board[15] != null && board[15] == board[16] && board[15] == board[17])
        ||(board[9] != null && board[9] == board[12] && board[9] == board[15])
        ||(board[10] != null && board[10] == board[13] && board[10] == board[16])
        ||(board[11] != null && board[11] == board[14] && board[11] == board[17])
        ||(board[9] != null && board[9] == board[13] && board[9] == board[17])
        ||(board[11] != null && board[11] == board[13] && board[11] == board[15])
        ||(board[18] != null && board[18] == board[19] && board[18] == board[20])
        ||(board[21] != null && board[21] == board[22] && board[21] == board[23])
        ||(board[24] != null && board[24] == board[25] && board[24] == board[26])
        ||(board[18] != null && board[18] == board[21] && board[18] == board[24])
        ||(board[19] != null && board[19] == board[22] && board[19] == board[25])
        ||(board[20] != null && board[20] == board[23] && board[20] == board[26])

```

```

||(board[18] != null && board[18] == board[22] && board[18] == board[26])
||(board[20] != null && board[20] == board[22] && board[20] == board[24])
||(board[0] != null && board[0] == board[13] && board[0] == board[26])
||(board[2] != null && board[2] == board[13] && board[2] == board[24]);
}

```

//menyemak samada semua kotak di penuhkan atau tidak

```

public boolean boardFilledUp() {
    for (int i = 0; i < board.length; i++) {
        if (board[i] == null) {
            return false;
        }
    }
    return true;
}

```

//bebenang permainan

```

public synchronized boolean legalMove(int location, Player player) {
    if (player == currentPlayer && board[location] == null) {
        board[location] = currentPlayer;
        currentPlayer = currentPlayer.opponent;
        currentPlayer.otherPlayerMoved(location);
        return true;
    }
    return false;
}

```

class Player extends Thread {

```

    char mark;
    Player opponent;
    Socket socket;
    BufferedReader input;
    PrintWriter output;

```

//penggunaan socket untuk mengawal permainan multi - pemain

```

public Player(Socket socket, char mark) {
    this.socket = socket;
    this.mark = mark;
    try {
        input = new BufferedReader(
            new InputStreamReader(socket.getInputStream()));
        output = new PrintWriter(socket.getOutputStream(), true);
        output.println("WELCOME " + mark);
        output.println("MESSAGE Waiting for opponent to connect");
    } catch (IOException e) {
        System.out.println("Player died: " + e);
    }
}

```



```
}  
}
```

```
public void setOpponent(Player opponent) {  
    this.opponent = opponent;  
}
```

```
//mengawal paparan mesej
```

```
public void otherPlayerMoved(int location) {  
    output.println("OPPONENT_MOVED " + location);  
    output.println(  
        hasWinner() ? "DEFEAT" : boardFilledUp() ? "TIE" : "");  
}
```

```
public void run() {  
    try {  
        //kesemua pemain telah dihubungkan  
        output.println("MESSAGE All players connected");  
  
        //memberitahu pemain pertama untuk jalan  
        if (mark == 'X') {  
            output.println("MESSAGE Your move");  
        }  
  
        while (true) {  
            String command = input.readLine();  
            if (command.startsWith("MOVE")) {  
                int location = Integer.parseInt(command.substring(5));  
                if (legalMove(location, this)) {  
                    output.println("VALID_MOVE");  
                    output.println(hasWinner() ? "VICTORY"  
                        : boardFilledUp() ? "TIE"  
                        : "");  
                } else {  
                    output.println("MESSAGE ?");  
                }  
            } else if (command.startsWith("QUIT")) {  
                return;  
            }  
        }  
    } catch (IOException e) {  
        System.out.println("Player died: " + e);  
    } finally {  
        try {socket.close();} catch (IOException e) {} } } }
```

8. Di bawah ini pula, kod program bagi pelanggan permainan.

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.io.InputStreamReader;
import java.io.BufferedReader;
import java.io.PrintWriter;
import java.io.IOException;
import java.net.Socket;
import java.net.ServerSocket;
```

```
/**
 * Pelanggan permainan untuk permainan 3D TTT
 * Client -> Server      Server -> Client
 * -----
 * MOVE <n> (0 <= n <= 26)  WELCOME <char> (char in {X, O})
 * QUIT                    VALID_MOVE
 *                          OTHER_PLAYER_MOVED <n>
 *                          VICTORY
 *                          DEFEAT
 *                          TIE
 *                          MESSAGE <text>
 */
```

```
public class TicTacToeClient {

    private JFrame frame = new JFrame("Tic Tac Toe");
    private JLabel messageLabel = new JLabel("");
    private ImageIcon icon;
    private ImageIcon opponentIcon;

    private Square[] board = new Square[27];
    private Square currentSquare;

    private static int PORT = 8901;
    private Socket socket;
    private BufferedReader in;
    private PrintWriter out;

    //memaparkan antaramuka setelah dihubungkan
    public TicTacToeClient(String serverAddress) throws Exception {
```

```

// Set rangkaian menggunakan socket
socket = new Socket(serverAddress, PORT);
in = new BufferedReader(new InputStreamReader(
    socket.getInputStream()));
out = new PrintWriter(socket.getOutputStream(), true);

// paparan antaramuka
messageLabel.setBackground(Color.lightGray);
frame.getContentPane().add(messageLabel, "South");

JPanel boardPanel = new JPanel();
boardPanel.setBackground(Color.black);
boardPanel.setLayout(new GridLayout(9,3,1,1));
for (int i = 0; i < board.length; i++) {
    final int j = i;
    board[i] = new Square();
    board[i].addMouseListener(new MouseAdapter() {
        public void mousePressed(MouseEvent e) {
            currentSquare = board[j];
            out.println("MOVE " + j);}});
    boardPanel.add(board[i]);
}
frame.getContentPane().add(boardPanel, "Center");
}

public void play() throws Exception {
    String response;
    try {
        response = in.readLine();
        if (response.startsWith("WELCOME")) {
            char mark = response.charAt(8);
            icon = new ImageIcon(mark == 'X' ? "ex.gif" : "ball.gif");
            opponentIcon = new ImageIcon(mark == 'X' ? "ball.gif" : "ex.gif");
            frame.setTitle("Tic Tac Toe - Player " + mark);
        }
        while (true) {
            response = in.readLine();
            if (response.startsWith("VALID_MOVE")) {
                messageLabel.setText("Valid move, please wait");
                currentSquare.setIcon(icon);
                currentSquare.repaint();
            } else if (response.startsWith("OPPONENT_MOVED")) {
                int loc = Integer.parseInt(response.substring(15));
                board[loc].setIcon(opponentIcon);
                board[loc].repaint();
            }
        }
    }
}

```



```

        messageLabel.setText("Opponent moved, your turn");
    } else if (response.startsWith("VICTORY")) {
        messageLabel.setText("You win");
        break;
    } else if (response.startsWith("DEFEAT")) {
        messageLabel.setText("You lose");
        break;
    } else if (response.startsWith("TIE")) {
        messageLabel.setText("You tied");
        break;
    } else if (response.startsWith("MESSAGE")) {
        messageLabel.setText(response.substring(8));
    }
}
out.println("QUIT");
}
finally {
    socket.close();
}
}

```

```

private boolean wantsToPlayAgain() {
    int response = JOptionPane.showConfirmDialog(frame,
        "Want to play again?",
        "Tic Tac Toe is Fun Fun Fun",
        JOptionPane.YES_NO_OPTION);
    frame.dispose();
    return response == JOptionPane.YES_OPTION;
}

```

```

static class Square extends JPanel {
    JLabel label = new JLabel((Icon)null);

    public Square() {
        setBackground(Color.yellow);
        add(label);
    }

    public void setIcon(Icon icon) {
        label.setIcon(icon);
    }
}

```

//larikan aplikasi pelanggan

```
public static void main(String[] args) throws Exception {  
    while (true) {  
        String serverAddress = (args.length == 0) ? "localhost" : args[1];  
        TicTacToeClient client = new TicTacToeClient(serverAddress);  
        client.frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        client.frame.setSize(300,500);  
        client.frame.show();  
        client.play();  
        if (!client.wantsToPlayAgain()) {  
            break;  
        }  
    }  
}
```

Rujukan

Buku:

Data Communication & Networking Second Edition McGraw Hill. Behrouz

A.Forouzan

Programming in Java. Sejana Publishing Dr P. Sellapan

H.M.Deitel & P.J.Deitel. Java How to Program. Prentice Hall, 1999

Chris Crawford: The Art Of Computer Design, Washington State University. 1997

Internet :

<http://netgames.indo.net.id/action/i-games.htm>

<http://java.sun.com/>

<http://www.minderackers.com>

<http://www.javaworld.com/channel-content/jw-applied-index.shtml>

<http://wwwj3d.org/>